

DEPARTMENT OF ELECTRONIC SYSTEMS

8TH SEMESTER

---

## Autonomous Hovering with a Quadrotor Helicopter

---

*Authors:*

Christian Fink PETERSEN  
Henrik HANSEN  
Steffen LARSSON  
Lars Bo THEILGAARD MADSEN  
Michael RIMESTAD

*Supervisor:*  
Martin KRAGELUND  
*Assist Supervisor:*  
Morten BISGAARD

June 2, 2008



**Title:**

Autonomous Hovering with a  
Quadrotor Helicopter

**Theme:**

Intelligent Autonomous Systems

**Project period:**

8th semester, spring 2008

**Project group:**

08gr833

**Members of group:**

Michael Rimestad  
Christian Fink Petersen  
Henrik Hansen  
Steffen Larsson  
Lars Bo Theilgaard Madsen

**Supervisor:**

Martin Kragelund  
Morten Bisgaard

**Copies:** 8

**Pages:** 126

**Attachment:** One CD

**Printed:** June 2nd, 2008

**Synopsis:**

The purpose of this project is to make a quadrotor helicopter (X-Pro) hover. In order to achieve this, the helicopter is first analysed to understand the physical perspectives. The model of the X-Pro is divided into three parts, motor/gear, rotor and body. An analysis of the electronics mounted on the X-Pro is made in order to achieve knowledge about the gyroscope and mixer.

The motors are modelled using system identification as ARX models. The rotors are found to be described by a second order polynomial. The X-Pro body is modelled as a rigid object using the dynamics and kinematics relationship.

The identified model of the X-Pro is linearised, in order to design a LQR controller. A PID controller is also designed. To track the angle and position of the X-Pro a Vicon Motion Tracking System is used as sensors to close the control loops. Both controllers are implemented and achieve the objective of hovering.



# Preface

This report is written by group 833 on the 8th semester at the Institute for Electronic Systems, in the period from February the 2nd to June the 3rd, 2008. The theme of the semester is "Modelling and Control". The purpose with the semester project is "To understand the concept of modern control", and "develop a model of a mechanical system to facilitate the application of modern control" [Gui07, p. 22].

The report is organized in 13 chapters, and an appendix. In the following a brief overview of all the chapters will be outlined, to give the reader insight into the overall structure and contents of the report.

- **Chapter 1, Introduction** Gives an introduction to the project, including setting up the objectives for the quadrotor helicopter

## Part I Analysis

- **Chapter 2, System Description** Gives a thorough system description of the X-Pro quadrotor helicopter, including the sensors and actuators made available for the project. Concludes with the requirements specification for the X-Pro.
- **Chapter 3, Requirement Specification** The list of requirements for autonomous flight in hover.

## Part I Modelling

- **Chapter 4-8, Modelling** The first 5 chapters under the modelling part describes and derives the model. The first chapter gives a general overview of the model of the X-Pro, which is basis for the following 4 chapters, that describes the Force Generation that affects the x-pro, the Projection of the forces followed by the model for the Dynamic and Kinematic part of the X-Pro.
- **Chapter 9, Combining the model** In this chapter the models derived in the first part of the modelling are combined to one.

- 
- **Chapter 10-11, Model Verification** The combined model is in this chapter compared to flight data from the real model, to verify if the combined model is responding correct according to the real model. And the linear model compared to the non-linear model.
  - **Chapter 12, Model Discussion** The model part discussion, wraps up the model part.

## Part II Controller

- **Chapter 13, PID Controller Design** This is the first chapter in the controller design. It describes the implementation and test of a simple PID controller controlling the X-Pro.
- **Chapter 14, States Space Controller** This chapter describes how a state space controller is designed using optimal control.
- **Chapter 15, Controller Discussion** The controller part discussion, wraps up the controller part

## Part II Project Delimitation

- **Chapter 16, Conclusion** This chapter makes a conclusion of the results in the project.

References to source material are indicated with reduced name and the year of publication and when known, the page number. Figures, tables and equations are referred to by the number of the object.

Bibliography and appendix can be found in the last part of the report and the cd can be found on the back page. The appendix contains information to support elements in the report, this includes an analysis of the existing hardware on the X-Pro, data logging for motor estimation, model linearisation, a description of the used Vicon Motion Tracking System and available Simulink blocks and last the calculation of the inertia tensor. Matlab files, source code, test results and electronic available literature, can be found on the CD marked with the relevant path on the CD.

### Project participant:

---

Michael Rimestad

---

Christian Fink Petersen

---

Henrik Hansen

---

Steffen Larsson

---

Lars Bo Theilgaard Madsen

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>I</b>	<b>Analysis</b>	<b>3</b>
<b>2</b>	<b>System Description</b>	<b>5</b>
2.1	The Design . . . . .	6
2.2	Basic Control of The X-Pro . . . . .	6
2.3	The R/C Transmitter . . . . .	8
2.4	System Blocks . . . . .	9
2.5	Signal Flow . . . . .	10
<b>3</b>	<b>Requirement Specification</b>	<b>13</b>
3.1	Accepttest Specification . . . . .	14
<b>II</b>	<b>Modelling</b>	<b>15</b>
<b>4</b>	<b>Modelling Overview</b>	<b>17</b>
4.1	Orientation . . . . .	18
<b>5</b>	<b>Force Generation</b>	<b>21</b>
5.1	Introduction . . . . .	21

## Contents

---

5.2 Motor and Gear . . . . .	21
5.3 Rotor . . . . .	31
<b>6 Projection</b>	<b>35</b>
6.1 Force . . . . .	35
6.2 Torque . . . . .	37
<b>7 Dynamics</b>	<b>39</b>
7.1 Forces to Accelerations . . . . .	39
7.2 Moment to Angular Acceleration . . . . .	40
<b>8 Kinematics</b>	<b>43</b>
8.1 Linear Acceleration . . . . .	43
8.2 Angular Velocity . . . . .	44
<b>9 Combining the Model</b>	<b>47</b>
<b>10 Non-Linear Model Verification</b>	<b>53</b>
10.1 Obtaining the test Data . . . . .	53
10.2 Collected Data . . . . .	54
10.3 Model Data . . . . .	55
<b>11 Linear Model Verification</b>	<b>57</b>
11.1 Throttle . . . . .	57
11.2 Roll . . . . .	58
11.3 Pitch . . . . .	59
11.4 Yaw . . . . .	59
<b>12 Model Discussion</b>	<b>61</b>

<b>III Controller</b>	<b>63</b>
<b>13 PID Controller</b>	<b>65</b>
13.1 Implementing the PID Controller . . . . .	65
<b>14 State Space Controller</b>	<b>71</b>
14.1 Control Law for the State Space Controller . . . . .	72
14.2 Introducing Reference Input . . . . .	74
14.3 Observer Design . . . . .	75
14.4 Combining Control Law and Observer . . . . .	76
14.5 Testing the LQR Controller . . . . .	77
<b>15 Controller Discussion</b>	<b>81</b>
<b>IV Closure</b>	<b>83</b>
<b>16 Conclusion</b>	<b>85</b>
<b>17 Perspective</b>	<b>87</b>
<b>Bibliography</b>	<b>89</b>
<b>V Appendix</b>	<b>91</b>
<b>A Analysis of the X-Pro hardware</b>	<b>93</b>
A.1 R/C Transmitter . . . . .	93
A.2 Mixer . . . . .	95
A.3 Gyroscope . . . . .	97
<b>B Data Logging for Motor Estimation</b>	<b>101</b>
B.1 Logging the Data . . . . .	101
B.2 Logging the PWM Signal . . . . .	102

## Contents

---

B.3 Logging the Revolutions . . . . .	102
<b>C Model Linearisation</b>	<b>105</b>
C.1 Method . . . . .	105
C.2 Operating Point . . . . .	106
C.3 Linearisation of the Body . . . . .	106
C.4 Linearisation of the Generated Forces . . . . .	112
C.5 Linearisation of the Generated Torque . . . . .	113
C.6 State Space Representation of the Complete Model . . . . .	114
<b>D Motion Tracking Laboratory</b>	<b>121</b>
<b>E Available Simulink Blocks</b>	<b>123</b>
E.1 ViconLink . . . . .	123
E.2 ConLink . . . . .	123
<b>F Calculating the Inertia tensor</b>	<b>125</b>

# Introduction

Unmanned aerial vehicles (UAV) are becoming increasingly popular, both in the private and the public sectors, especially for surveillance assignments of locations hard to access in a conventional manner. Helicopters have an advantage over aeroplanes as they can manoeuvre on small spaces and do vertical take off and landing. This project deals with a quadrotor helicopter, which differs from ordinary helicopters by having 4 horizontal rotors and no vertical rotors. The quad rotor helicopter offers the same functionality as an ordinary helicopter in the form of attitude and altitude control.

Furthermore, flying UAV's often requires a lot of human interaction, using a remote control transmitter. A more interesting approach is autonomous flight, reducing the dependency on human interaction and visual contact with the UAV, enhancing the potential of them. This could be by helping the human to keep the helicopter more steady, which would be preferable if the UAV for instance had a camera mounted.

One of the biggest problems with a UAV is to keep track of its position and orientation. This will not be the focus in this project as a motion lab are made available for the project group, which can track the helicopter's position and orientation.

The purpose of this project is to get the quadrotor helicopter to autonomous hover.



## Part I

# Analysis



# Chapter 2

## System Description

The quadrotor helicopter used in this project is a Draganflyer X-Pro (X-Pro). It differs from traditional helicopters regarding the design as it has 4 horizontal rotors and no vertical rotors. Traditional helicopters can adjust the angle of the rotor blades as well in order to control the helicopter, but on the X-Pro the blades have a static angle. The only variable that can be adjusted in flight is the rotational speed of the rotors. In figure 2.1 the X-Pro is mounted on a stand.



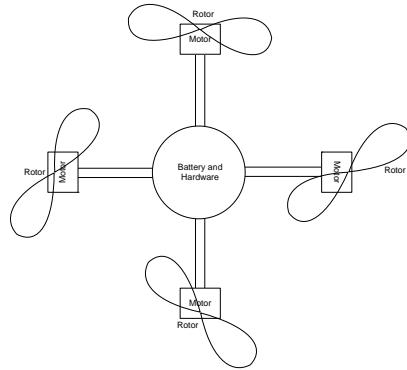
**Figure 2.1:** The Dragan Flyer X-Pro mounted on a stand.

The reason for this is to aid the development of a model and controllers for the X-Pro, without the risk of flying into the wall. This will of course give reduce the movements of the X-Pro, but

it will make it easier to test if the input-output relationship are as expected.

## 2.1 The Design

The X-Pro consists of a base in the center and 4 arms equally spaced in a shape of a cross. The motors are mounted on the end of the arms and connected to the rotors using 1 to 10 gearing. The base of the X-Pro is the joint between the 4 arms and contains a battery and the remote control (R/C) electronics. The basic structure of the X-Pro can be seen in figure 2.2.



**Figure 2.2:** A sketch of the main body items seen from above.

In the major part of the project the X-Pro will be mounted on the stand and due to that, the battery will be unmounted and an external power supply will be connected instead.

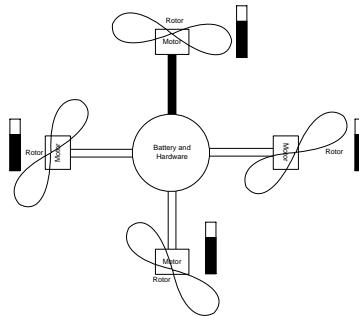
## 2.2 Basic Control of The X-Pro

The X-Pro is controlled by alternating the rotational velocity of the rotors. The front and back rotors are rotating counter clockwise and the rotors to the left and right are rotating clockwise. A mixer is part of the electronics on the X-Pro. It handles the incoming control signals in the form of throttle, yaw, pitch and roll and translates them into motor control signals. Yaw is also known as the heading. The mixer makes it possible to control the X-Pro like an traditional helicopter. In the following these terms will be described supported with figures that illustrates the X-Pro, including bars to show the level of motor speed. A full black bar indicates full motor speed and white box is zero motor speed. With all 4 bars at 50% the X-Pro will hover at a fixed position. In all figures the X-Pro will be seen from above and with one black arm to mark the front.

### 2.2.1 Throttle

When throttle is increased the revolution on the four motors increases simultaneously. The rotors will then create a lift and the X-Pro will increase in altitude. In figure 2.3 the bars indicate that

the rotational speed is above average. The X-Pro is seen from above and the black bar is the

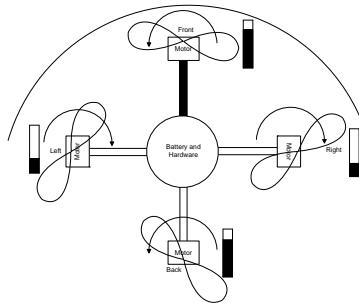


**Figure 2.3:** Illustration of the X-Pro with increased throttle which results in increased altitude.

front.

### 2.2.2 Yaw

When the heading is changed the overall lift is maintained, although the speed of the rotors changes. To make the X-Pro rotate clockwise the speed of the left and right rotors are decreased and the speed on the front- and back rotors are increased. This way the X-Pro maintains the same lift force. Furthermore, the X-Pro increases the drag force from the two rotors, with the highest velocity, which makes the X-Pro rotate. An illustration of the X-Pro turning clockwise can be seen in figure 2.4. To make it turn counter clockwise the opposite is obviously applied.

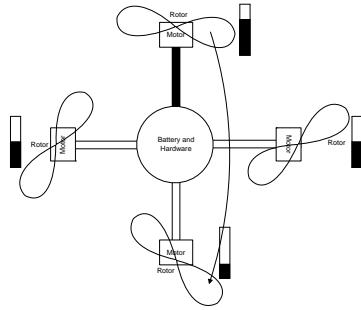


**Figure 2.4:** Illustration of the change of rotor speed needed to turn the X-Pro clockwise.

### 2.2.3 Pitch

Pitch is used in order to make the X-Pro move forward and backwards. Moving forward is a result of tilting the X-Pro forward, where the velocity of the front rotor is decreased, and the back rotor is increased. During this movement the total lift needs to be maintained, keeping the velocity of the side rotors unchanged. Figure 2.5 illustrates the X-Pro pitching backwards.

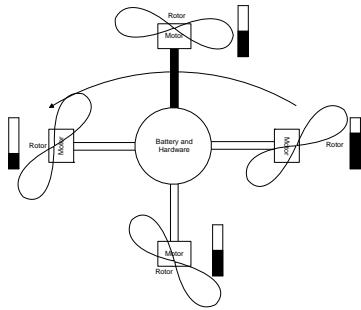
The opposite happens when tilting the X-Pro backwards.



**Figure 2.5:** Illustration of the changing rotor speed, needed to tilt the X-Pro backwards.

#### 2.2.4 Roll

Roll is used to make the X-Pro move sideways. Roll is performed like pitch, except that the velocity of the side rotors are changed, instead of the front- and back rotors. The roll principle is illustrated in figure 2.6.



**Figure 2.6:** Illustration of change of rotor speed needed to make the X-Pro roll to the left.

### 2.3 The R/C Transmitter

The X-Pro is controlled using an ordinary R/C transmitter. The R/C transmitter used in this project is a Futaba T9CAP with 9 channels and can be seen in figure 2.7. As mentioned in previously sections only 4 signals are to be controlled, therefor only 4 channels are going to be used.

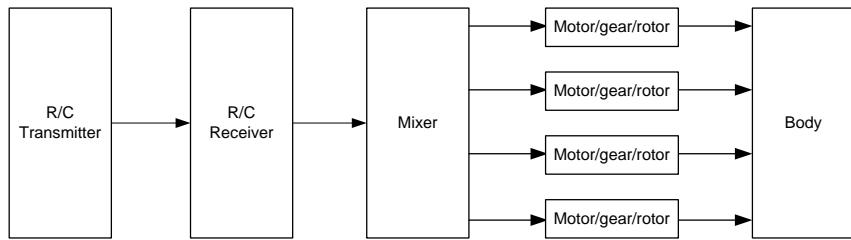
To control the 4 signals, 2 sticks on the R/C transmitter are used. The left stick controls yaw and throttle and the right stick the roll and pitch. When moving the left stick vertically, the throttle signal is manipulated. When the stick is in down position, zero throttle is send. When the stick is in upper position, full throttle is sent. Moving the left stick horizontally, the yaw signal is manipulated. Yaw is zero when the stick is in the center, and when it is moved to the left, the X-Pro turns counterclockwise. The rotation is clockwise, when the stick is moved to the right. Centering the right stick means that the R/C transmitter sends zero roll and pitch. Moving the stick vertically, pitch is manipulated and horizontally roll is manipulated. Combining these four control signals, it is possible to move the X-Pro, with six degrees of freedom.



*Figure 2.7: Futaba T9CAP radio controller used for controlling the X-Pro.*

## 2.4 System Blocks

The overall system has been described, and the system is now divided into blocks, in order to make an analysis. A block diagram of the system blocks is illustrated in figure 2.8.



*Figure 2.8: Blocks in the system.*

### 2.4.1 R/C Transmitter and Receiver

The first block is the R/C transmitter, that sends 4 control signals (throttle, yaw, pitch and roll). This is sent to the R/C receiver, that passes the signals to the X-Pro mixer. The data is sent as pulse signals. The amount of control is defined by the ON time of the pulse, which can vary from 1 ms to 2 ms as described in appendix A.

### 2.4.2 Mixer

Increasing or decreasing throttle or yaw signal will affect all the motors, while pitch and roll only affects 2 of the motors. The input received by the X-Pro is the only information about throttle, pitch, yaw and roll. The mixer translates this information into motor control signals, which is further described in appendix A.

### 2.4.3 Motor/gear/rotor

This block consists of four brushed DC-motors and takes PWM<sup>1</sup> signals as input. The PWM signals have a constant voltage and only the duty cycle varies depending on what is sent. As mentioned in the previously section the front and back motors are mounted in such a way that the shafts are rotating counter clockwise and the left and right are rotating clockwise.

The gear on each motor consists of two toothed wheels connected with a belt. The toothed wheels on the motors are 10 times smaller than the wheel mounted on the rotor. This ratio results in a larger torque but reduces the velocity of the rotors.

The rotors are mounted directly on the gears and each of them consists of two blades pointing in opposing directions. The angle of the blades are static, and the lift they produce, therefore only dependent on the velocity of the motors. The drag that each rotor produces, depends on the rotational direction of the respective rotor and velocity.

### 2.4.4 Gyroscope

A gyroscope is mounted on the X-Pro, which measures angular velocities. This is described in details in appendix A. Through an internal feedback loop the gyroscope regulates the rotational speed of the rotors. The gyroscope assists in stabilizing the X-Pro by countering any unintended fast movement.

## 2.5 Signal Flow

Setting up the R/C transmitter it is important to know the signal flow. The complete system consists of an Interface PC for the Vicon Motion Tracking System and for the R/C transmitter. Another computer, the Controller PC, is used to run all the calculations for the Simulink controller to the X-Pro. The system is shown in figure 2.9.

The different elements in the system will be described in the following sections and a description of the X-Pro hardware can be found in appendix A.

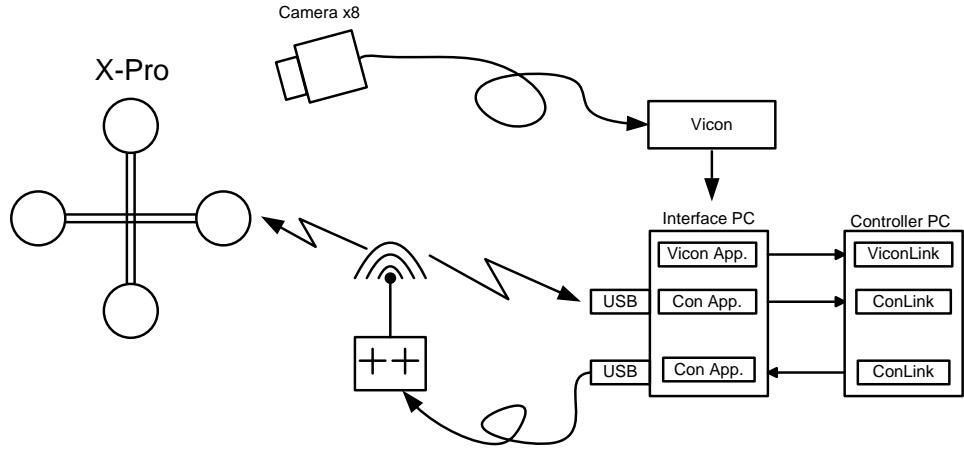
### 2.5.1 Vicon Motion Tracking System

One of the available elements in the system is the Vicon Motion Tracking System which is able to track an object, using infrared light reflecting off small spheres mounted on the object. By tracking the sphere via 8 cameras, the system is able to calculate the position and Euler angles of the object. A more elaborate analysis of the Motion Tracking System can be seen in appendix D.

The interface PC uses the ViconIQ application to gather the measurements from the Vicon Motion Tracking System. Any object can be tracked online at a refresh rate at 100 Hz and the

---

<sup>1</sup>Pulse Width Modulation



**Figure 2.9:** Illustration of the complete setup and the signal flow.

parameters are sent via the Controller Link application to the Controller PC.

The Controller PC includes the ViconLink block which is able to gather the Vicon parameters at the same refresh rate over a TCP/IP connection. The ViconLink block is described in appendix E.

### 2.5.2 Control Signal Flow

The control signal is the signal that makes it possible to manipulate the X-Pro. The control signal consists of 4 parameters representing throttle, yaw, pitch and roll as mentioned in section 2.2.

The first step is the R/C transmitter that sends an RF signal which is received by an R/C receiver on the X-Pro and the Interface PC. The Interface PC uses an available ConLink application to send the control signal to the Controller PC. The Controller PC then uses a ConLink Simulink block to receive the control signal. The ConLink block also includes the opportunity to send a control signal the other way around which in turn makes it possible to control the X-Pro from the Controller PC. A more detailed description of the ConLink simulink block can be found in appendix E.

From the Controller PC it has been observed, that the values sent by the ConLink block differs from the received values. A mapping must be done in order to be able to switch between flying with the R/C transmitter and using a Simulink controller on the Controller PC. The mapping will include transfer functions, that compensate for this problem. The transfer functions are derived in appendix E.



# Chapter 3

## Requirement Specification

The project proposal is about modelling and controlling the quadrotor helicopter flying in hover. Thus, the first step is to model the different parts of the X-Pro in order to obtain a complete model. As a quadrotor helicopter is affected by a variety of forces, part of the task also consists of disregarding effects, that only have minor influence on the X-Pro. Otherwise the model will get too complex. The model of the X-Pro is then validated based on flight data from the X-Pro.

The next step after model development is to design a controller based on the X-Pro model. To get a linearised controller the model needs to be linearised around the equilibrium point for hover, which needs to be identified. Two types of controllers for the X-Pro is designed. The first is a simple PID controller, which can be designed by trial and error. The other is a MIMO controller based on the linearised model. To obtain the aim of autonomous flight in hover, the attitude and position of the X-Pro needs to be controlled.

As it is not expected to maintain an exact attitude and position during flight, caused by external disturbances, a set of success requirements for autonomous flight in hover are outlined :

1. The z axis positng should be kept within  $\pm 0.05$  m.
2. The x and y axis the positng should be kept within  $\pm 0.1$  m.
3. The angle around the z axis should be kept within  $\pm 0.1$  radian.
4. When changing the position 0.5 meter in either the x or y axis, the X-Pro should move and place itself and be stable in the new position within 2 seconds.

The angle around the x and y axis does not have any requirements as these change whenever the X-Pro is adjusting pitch and roll to maintain the position.

### 3.1 Accepttest Specification

To test the four requirements outlined above an accepttest specification is needed. This specification explains how these requirements can be tested. For these tests the position and attitude from the Vicon Motion Tracking System are used. Requirement one, two and tree can be tested at the same time, while requirement four has to be tested by independently.

#### Test 1: Keeping position and angle

In this test is is verified if the controller is able to keep the position and yaw angle of the X-Pro in autonomous hover.

1. The X-Pro is started and brought into hover position.
2. From the point where the X-Pro is in hover, the attitude and position of the X-Pro should be recorded.
3. After 10 seconds of flight in hover, the recording can stop and the X-Pro is brought back to the ground.
4. The attitude and position data are separated and the position and yaw angle are checked against their given requirements.

#### Test 2: Changing position

This test is used to verify if the controller is able to change the position of the X-Pro and if its able to stabilise the X-Pro within the required time.

1. The X-Pro is started and brought into hover position.
2. From the point where the X-Pro is in hover, the attitude and position of the X-Pro is recorded.
3. The reference for the x position or the y position are changed with  $\pm 0.5$  meter.
4. After 2 seconds of flight in hover in the new position, the recording can stop and the X-Pro is brought back to the ground.
5. The attitude and position data are separated and the position is checked against the given requirements.

# Part II

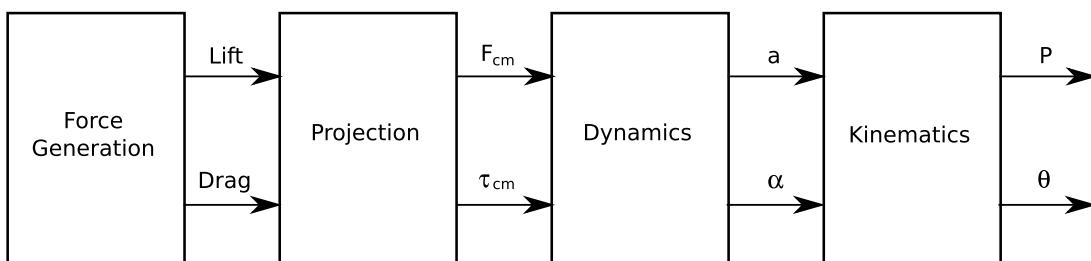
# Modelling



# Chapter 4

## Modelling Overview

In this part of the report, the model for the entire X-pro will be derived. The model will be divided up into four parts which are shown in figure 4.1. Furthermore the figure shows the interaction between the parts.



*Figure 4.1: Block diagram of the X-Pro model.*

A short description of the 4 parts follows:

- **Force Generation**

The Force Generation is a part that consists of the 3 elements, motors, gear and rotor. The result of the 3 elements will be 4 lift forces and 4 drag forces.

- **Projection**

Projection translates the lift and drag forces affecting the body to a total force and a torque vector. In the projection part, the external forces and torques affecting the X-Pro is also discussed.

- **Dynamics**

The Dynamics translates the total force and torque to an linear acceleration and angular acceleration of the body.

- **Kinematics**

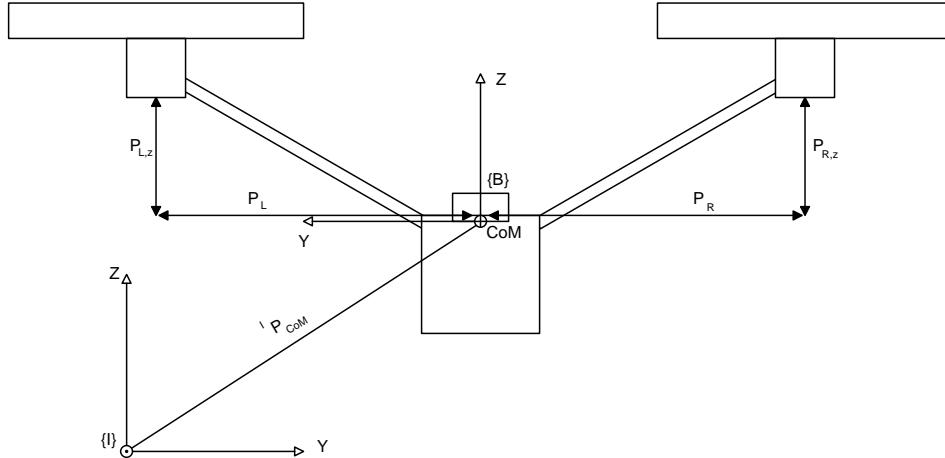
The Kinematics translates the accelerations to Euler angles and a position of the X-pro.

Combining the parts, a complete mathematical description will be derived. From this joint formula a controller for the X-Pro can be designed.

## 4.1 Orientation

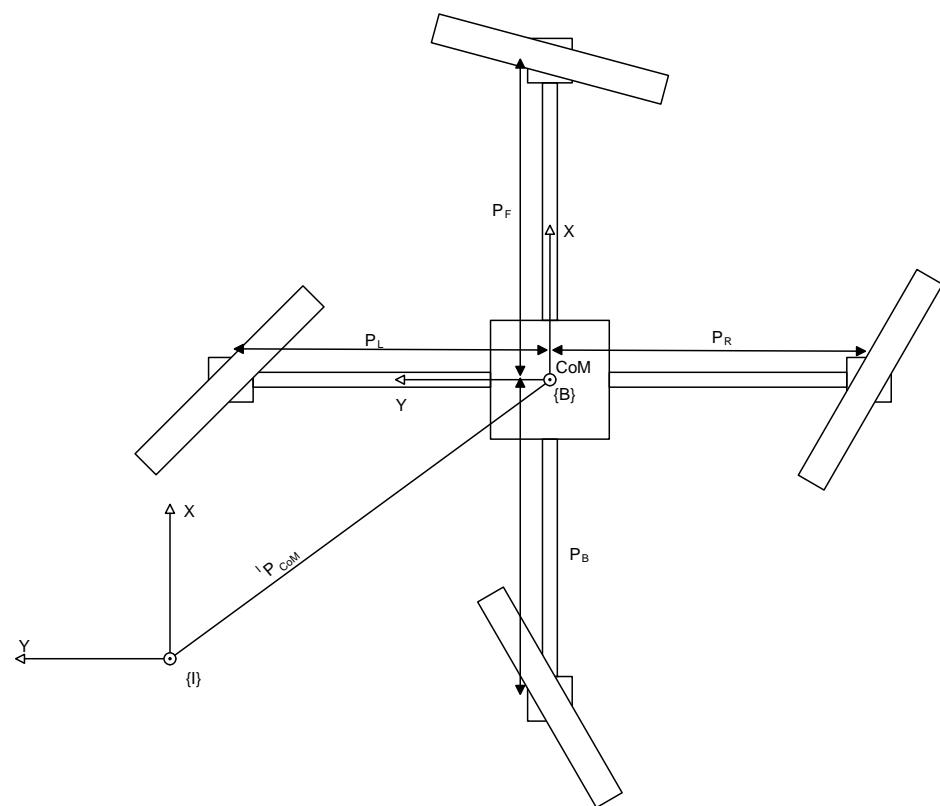
A part of the modelling is clarifying the orientation of the X-Pro. To do this 2 coordinate system will be used. One that describes the universe the X-Pro is flying in, and another placed in the center of the X-Pro. The universal coordinate system can placed freely on a surface that is non accelerating compared to the X-Pro. For instance in this case the universal coordinate system is placed on the ground. In order to identify which direction the base of the X-Pro is orientated, the coordinate system is placed in the center of mass of the X-Pro.

The placement of the coordinate systems has been outlined in figure 4.2 where the X-Pro is seen from behind and in figure 4.3 where it is seen from above. Each of the coordinate systems has a reference frame index - and will be referred to as a frame. The base coordinate system B will always be described relative to the universe coordinate system I. A frame is a set of 4 vectors describing the giving position and orientation. A description of a frame related to the X-Pro, could be a position vector and a rotation matrix.



**Figure 4.2:** The X-Pro from behind, with the coordinate systems in the base and the universe inserted.

The first part to be modelled is the force generators, which is done in the following chapter.

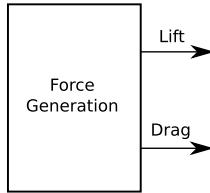


*Figure 4.3: The X-Pro from above with the coordinate systems inserted.*



# Chapter 5

## Force Generation



*Figure 5.1: Model for the Force generation.*

### 5.1 Introduction

The force generation chapter contains the parts of the X-Pro, that generates the forces affecting the position and the attitude. The block is illustrated in figure 5.1. It contains a section describing the rotor and a section describing the motor and gear. The sections have been made simultaneously as these parts are tightly coupled, and requires some of the results being used in both sections. The purpose of this chapter is to derive the correlation between the input signals and the force generated affecting the behavior of the body of the X-Pro.

### 5.2 Motor and Gear

In this section the model of the motor including gear will be discussed. The separation of the models for motor/gear and the rotor is due to some restrictions in the opportunities of measuring the lift and drag from the rotors. This way the dynamics will be kept in the motor/gear model, as measurements done on this part of the system can be calculated by computers, while the forces and torques from the rotors must be manually read.

A motor is a process that can be described by an electrical and a mechanical part. The process includes some parameters which can be measured and calculated by experiments. It is also known that this process includes two kinds of friction. A viscous friction that is known to be almost

linear and a dry friction that are non-linear. Furthermore it is decided to model the motor/gear with the rotor attached. As the drag from the rotor will be included in this model it will become non-linear.

To overcome the non-linearity problem some assumptions will be made. From the requirement specification in chapter 3 the main objective for the project is to make the X-Pro able to hover. In that case the model for the motor/gear will be based on data measured close to the rotational velocity where the X-Pro will hover. Parallel to modelling the motor, the model for the rotor will be made. From this work the lift from the rotor will be derived, as a function of the rotational velocity. In this chapter, information about the lift will be used to calculate the needed rotational velocity of the rotors, in order to make the X-Pro hover. This velocity will be used as the equilibrium point that will be used for a linear model for the motor/gear.

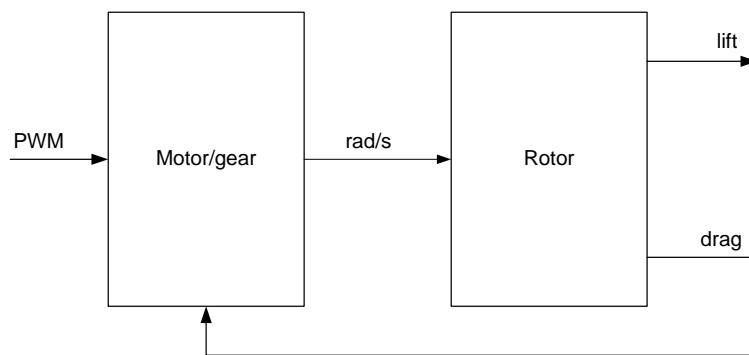
It is decided to make the model of the motor/gear using system identification (SID), as it can be difficult to derive all the equations mathematically with the gear and rotor attached to the motor.

First the general approach on system identification will be described. Then the method that are used in system identification is described. Then logging the data for the estimation part will be done, and lastly the estimation the model parameters is derived.

### 5.2.1 System Identification of Motor incl. Gear

To make a mathematical description of the motor including the gear, system identification will be used. To verify the model, a data set different from the one used for the system identification, will be used. System identification will be based on output reaction due to a specified input.

System identification is a way to make a transfer function that describes the input/output relationship. To get an overview of the processes and the inputs and outputs, a block diagram of the motor/gear and rotor has been made and can be seen in figure 5.2. The block described in this chapter is the motor/gear, where inputs are PWM signal and the outputs are the rotational velocity of the rotor. Because the rotor is attached to the motor during the SID, the drag from the rotor will contribute with some resistance. This is also illustrated on the figure.



**Figure 5.2:** Block diagram of the motor including the gear and the rotor.

The input signal which is a PWM signal, has the low level at 0 V and the high level at 12 V and constant frequency at 196 Hz. The resulting voltage on the motor is the mean of the voltage

over time which means that the voltage on the motor can be varied by the duty cycle only.

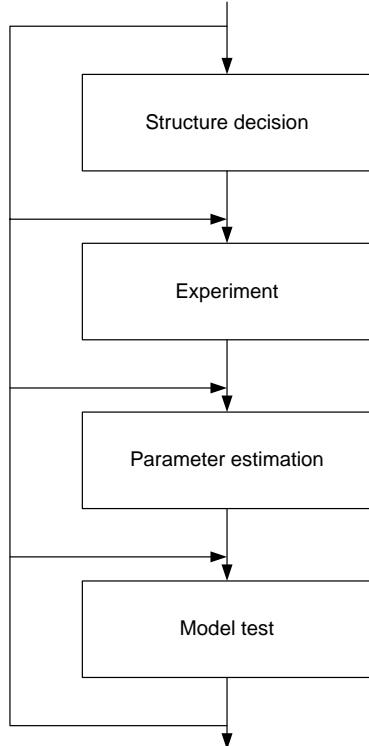
The output signal is the revolution on the gear measured in radians per second (rad/s). From this the transfer function can be expressed as

$$G = \frac{\text{gear revolutions}}{\text{dutycycle}} \left[ \frac{\text{rad/s}}{\text{PWM}} \right] \quad (5.1)$$

With a general description of the transfer function, system identification will be used in order to identify the function.

### 5.2.2 Steps in System Identification

System identification is often an iterative process as seen in figure 5.3. That is because it can be necessary to go back and adjust some parameters etc.



*Figure 5.3: Steps in system identification.*

A model structure is the first thing that needs to be chosen. Choosing a structure can be done on basis of knowledge of noise aspects and physical behavior. For instance if noise appears at the input channel, a certain model might be more preferable to choose than others. The same thing applies to systems where the knowledge of the system indicates that the model order is very high.

After this experiment, design must be done in order to get the input output relationship. This experiment is described in appendix B.

The parameters are typically estimated using functions in system identification toolbox in Matlab. Model validation is the final step, and is done in order to verify the result. This can be done with data from input- output measurements, and if the result is acceptable the model is verified. Otherwise it is needed to go back and adjust some parameters.

The strategy in choosing the model is to start with a simple linear e.g. first order ARX model, and then if it is necessary, iteratively approximate a more suitable one. First it is necessary to have a little introduction in model structures to choose between.

### 5.2.3 Structure decision

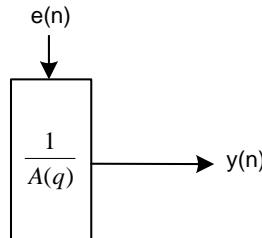
There are several structures to choose between in system identification. The models are based on following:

- AR (Auto Regressive)
- MA (Moving average)
- X (eXogenous)

The AR part is the linear regression of the current value of the series against one or more prior values of the series, and are given by:

$$A(q)y(n) = e(n) \quad (5.2)$$

Figure 5.4 shows an AR model of a system, and it can be seen that a signal can be reconstructed if only the white noise and the  $A(q)$  coefficients are known.



*Figure 5.4: AR model.*

MA is the moving average model, which represents a series of data generated by performing linear regression on white noise. It makes a mean of a number of samples. By rewriting equation (5.2) it is possible to go from an AR to a MA and vice versa, which the following equation shows.

$$y(n) = A(q)e(n) \quad (5.3)$$

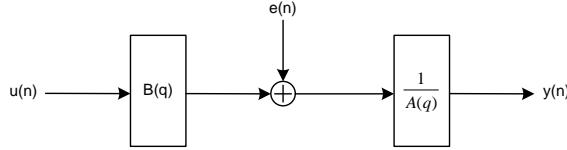
The polynomial  $A(q)$  is

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + \cdots + a_n q^{-n_a} \quad (5.4)$$

where  $n_a$  is the model order of the system,  $y(n)$  is the output of the model and  $e(n)$  is zero mean white noise, or the disturbance.  $q^{-1}$  is a back shift operator defined by

$$q^{-1}x(n) = x(n - 1) \quad (5.5)$$

The X part is eXogenous, which is included when having an input to the system. For instance the ARX (Auto Regressive eXogenous) model which can be seen in figure 5.5.



**Figure 5.5:** Block diagram of the ARX model.

The ARX model is expressed as

$$A(q)y(n) = B(q)u(n) + e(n) \quad (5.6)$$

where polynomial  $B(q)$  is

$$B(q) = b_0 + b_1q^{-1} + \dots + b_{n_b-1}q^{-(n_b-1)} \quad (5.7)$$

and  $n_b$  is the model order. The experiment block in figure 5.3 will be made under the assumption that it is in open loop since the X-Pro will be forced to be in a fixed position. Therefore no feedback from the gyroscopes are taken into consideration. Choosing the model order will be done by investigation the estimation data's spectral plot.

The ARX structure is a useful method to begin with for modelling the motor and gear. Disturbances in the form of white Gaussian noise is the only one expected. The ARMAX model, which can handle disturbances such as wind gust, might be a more interesting choice when flying the X-Pro outdoor. As mentioned in the introduction a linear model needs to be derived from a equilibrium point and the strategy is to start out with a simple model such as an ARX(2,1). E.g. an AR model with

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} \quad (5.8)$$

and

$$B(q) = b_0 + b_1q^{-1} \quad (5.9)$$

with one input. At this point the method for estimating the model has been clarified and the ARX model has been chosen as a model for the process. Calculating the equilibrium point provides the area of interest for the required data for parameter estimation.

### 5.2.4 Equilibrium point

In order to end up with a linearized controller, the motor needs to be linearized around an equilibrium point. Since the aim is to have the X-Pro hover, a optimal equilibrium point would be the angular velocity which creates just enough lift to overcome the gravitational force. This point is found by measuring the weight of the X-Pro and then using the correlation between force creation and rotational velocity found in section 5.3. The weight of the X-Pro without the battery is 1990 gram. As the X-Pro is to be flown indoors during this project, connected to a power supply, the weight of the battery is irrelevant. The necessary lift force required for each of the 4 rotors is then

$$F = \frac{9.82 \cdot 1.990}{4} = 4.885 \quad [\text{N}] \quad (5.10)$$

Using equation (5.16) from section 5.3 to calculate the necessary angular velocity required to generate this force

$$F_{Lift}(\Omega) = 0.2263 \cdot 10^{-3} \cdot \Omega^2 - 2.488 \cdot 10^{-3} \cdot \Omega + 11.97 \cdot 10^{-3} \quad [\text{N}] \quad (5.11)$$

Solving this equation for  $\Omega$  gives

$$\Omega = 152.35 \quad \left[ \frac{\text{rad}}{\text{s}} \right] \quad (5.12)$$

This gives an equilibrium point at 152.35 rad per seconds which is used to choose the experiment data to be used for parameter estimation.

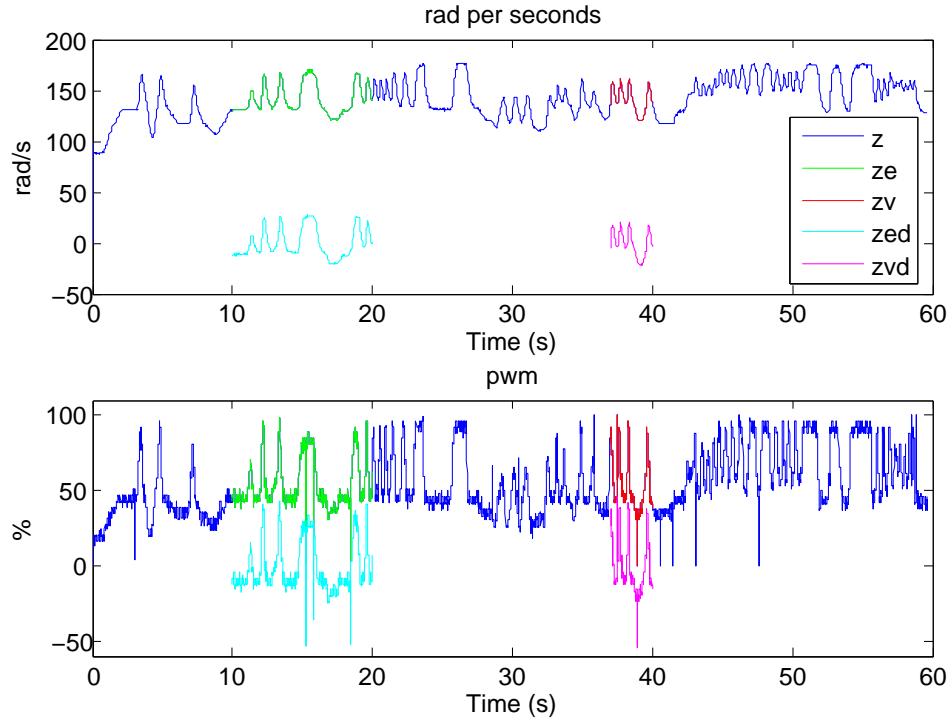
### 5.2.5 Experiment

Input and output was measured as described in appendix B at a frequency of 100 Hz in a time period at 60 seconds. Two data sets are picked out, one for estimation and one for validation. The sets are picked out in a matter so they vary around the estimated equilibrium point. Before the data sets can be used for estimating the parameters the trends in the signals are removed. The signals measured in the laboratory are named  $z$ . The set for estimation are named  $ze$  and the set for validation  $zv$ . The trends in the two sets are removed by the Matlab command, `detrend`, and will be named  $zed$  and  $zvd$  respectively. The signal can be seen in figure 5.6.

Detrending in this particular case removes the mean from signal, thus resulting in a signal without offset. Therefore the offset must be added to the transfer function in the end. With this in mind, the signals are prepared for parameter estimation.

### 5.2.6 Parameter Estimation

To estimate a model for the motor including the gear and with a rotor attached, the procedure described previously in section 5.2.2 will be used. The estimation will be done with the



**Figure 5.6:** Measured output and input. Output is the angular velocity and input is the duty cycle of the PWM signal.

use of Matlab and the m-file which is used throughout the estimation can be found on the CD\models\motor\motorgear.m.

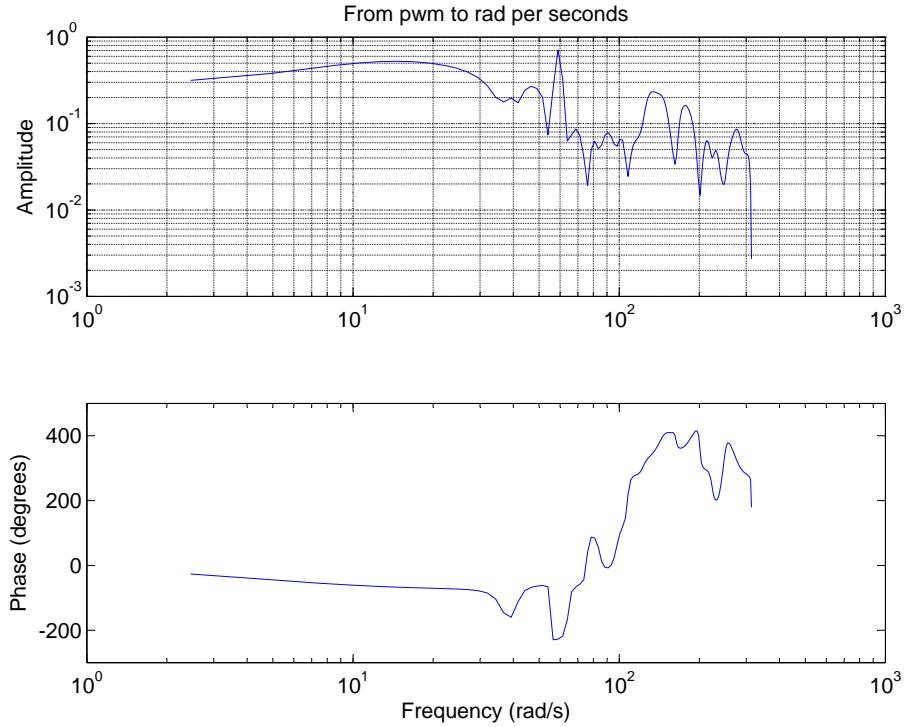
### Deciding the Model Order

The model is based on an ARX model and to decide the order of the model a spectral analysis is done. The spectral analysis represents the frequencies in the signal and can be seen in figure 5.7.

From the spectral analysis it can be seen that the amplitude is decreasing by 2 decades per 1 decade in frequency. This results in a second order model and using an ARX model the model will be a ARX(2,2).

### 5.2.7 Model Test

The ARX model with the estimated parameters has been compared to the detrended validation data,  $zvd$ . To compare the model with the estimated parameters, the command `compare` in Matlab has been used. The command computes an estimated output signal  $\hat{y}$  by the use of the ARX model and an actual input signal. The measured output signal is compared to  $\hat{y}$  and a fit is computed by



**Figure 5.7:** Spectral analysis of the measured signal.

$$\text{fit} = 100 \frac{1 - \text{norm}(\hat{y} - y)}{\text{norm}(y - \bar{y})} \quad (5.13)$$

where  $\bar{y}$  is the mean of  $y$ . In figure 5.8 the comparison of the signals can be seen.

From equation (5.13) the ARX model fit has been calculated to be 89.47 % and with the fact in mind that the signals include noise the fit is accepted.

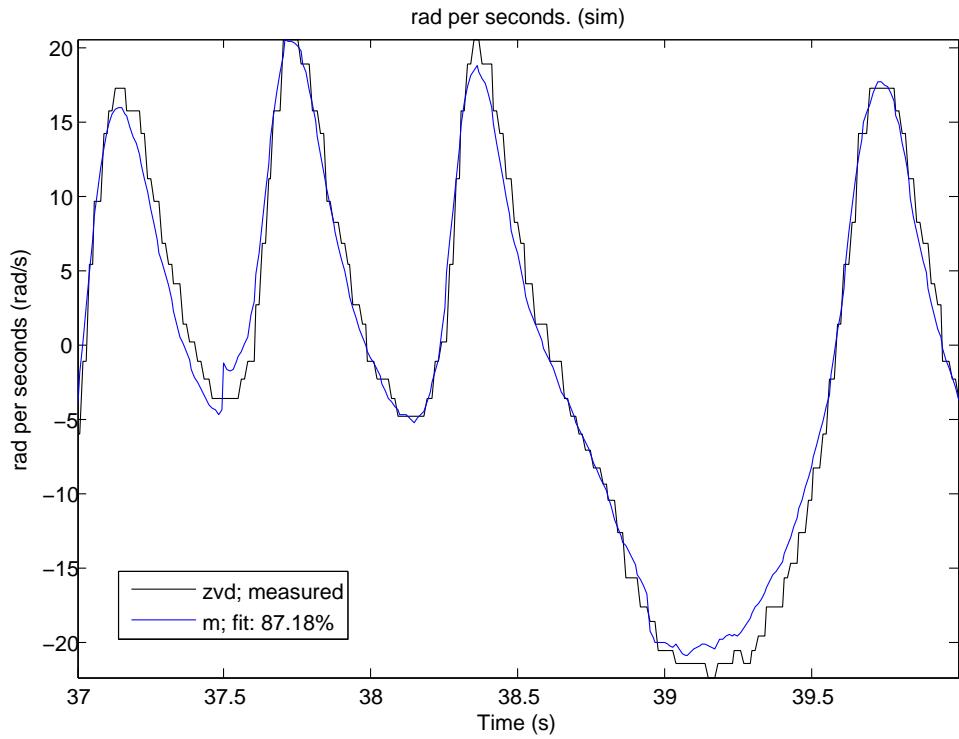
### Extracting the Transfer Function

When estimating the ARX model the parameters for the transfer function are also estimated. The transfer function, which is a discrete function, from the PWM signal to the revolutions in rad/s is then:

$$G(z) = \frac{0.0433z - 0.0227}{z^2 - 1.6306z + 0.6496} \quad \left[ \frac{\text{rad/s}}{\text{PWM}} \right] \quad (5.14)$$

Furthermore the offset mentioned in section 5.2.5 must be added to complete the model for the motor/gear. The offset has been calculated to be 85,79 rad/s.

A model that describes the process of a motor with gear and rotor attached are now derived. The model will be used in controller design.

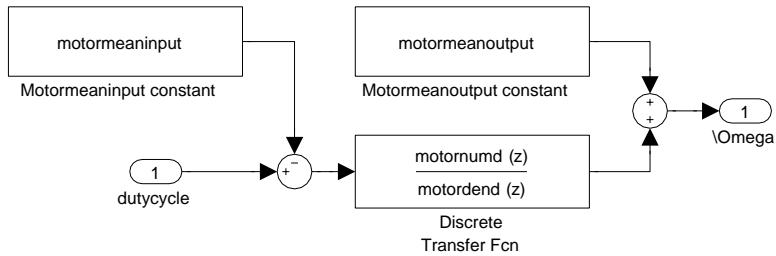


*Figure 5.8: ARX models compared to an actual output signal.*

### 5.2.8 Simulink Implementation

To complete the entire model of the X-Pro all elements are modelled and implemented in Simulink. The model for the motor/gear will be implemented in Simulink in the following section. The model will be implemented as a discrete function in the z-domain. From the m-file used to estimated the model, the numerator and denominator will be extracted.

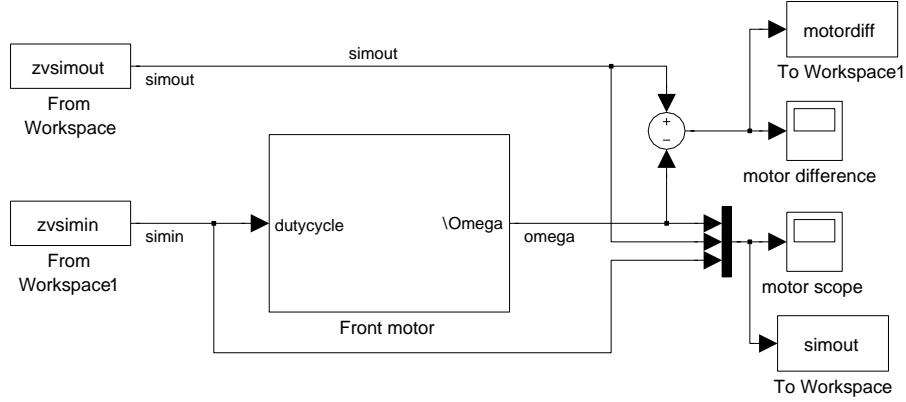
During the parameter estimation the measured input and output signals where detrended. This means that the mean of the signals was subtracted. To compensate to that the means must be added in the model. The addition of the offset means the at zero input the output will be equal to the offset value. The implemented model can be see in figure 5.9.



*Figure 5.9: Implemented motor model.*

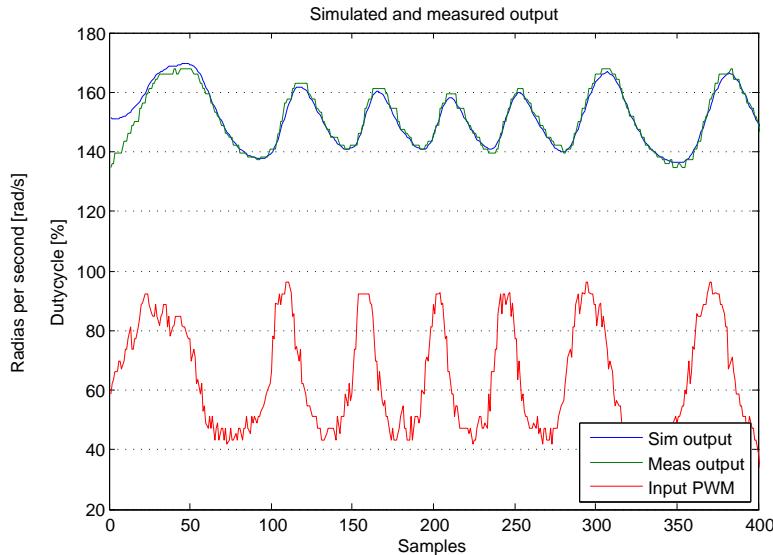
### 5.2.9 Model verification

The input to the model is the measured PWM signal. The mean of the input signal is subtracted to match the sequence used in the parameter estimation, in figure 5.6. The discrete transfer function calculates an output to which the mean of the output sequence is added. The Simulink model used for validation of the motor/gear model can be seen in figure 5.10.



**Figure 5.10:** Model used for validation of the implemented Simulink model of the motor/gear.

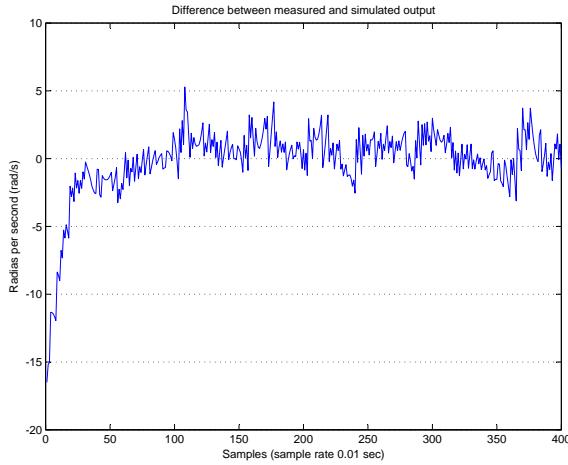
The model output is compared to the measured output signal. The signals can be seen in figure 5.11.



**Figure 5.11:** Plot of measured input and output and simulated output.

The implemented model computes an acceptable output compared to an actual measured output signal. Furthermore the simulated output have been subtracted from the measured output to seen the difference between the 2 signals. The result can be seen in figure 5.12.

It is assumed that the remaining variations in the difference are due to noise in the measured input and output signals. Hence the model of the motor/gear is accepted.



**Figure 5.12:** Plot of the difference between the simulated and measured output.

## 5.3 Rotor

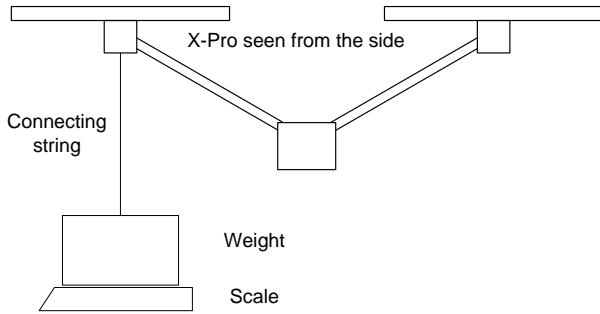
The rotor model describes how the rotations provided by the motor translates into the 2 forces, lift and drag. In order to push air downwards by rotational movement, the rotor blade needs to be tilted, which also results in air being pushed in a circular motion, creating lift and drag respectively. These forces are affected by the air, by the rotor, by the position of X-Pro and how it is tilted, thus making the derivation of a mathematical model complex. Limiting the model to consider positions and attitudes close to hover simplifies the model considerably. The resulting model can be described by a second-order polynomial taking the rotational speed as input and providing lift and drag as output. That the model is a second order polynomial is shown by the quad-rotor group from 2006, and is also indicated by how the following measurements fit the function [06g06, p. 151]. In order to identify the parameters for this model, a series of measurements are conducted.

### 5.3.1 Measuring lift

Measuring lift requires the X-Pro to be stabilized around hover without having any fixation for pitch. I.e. the arm with the rotor used for measuring can move freely vertically while all other degrees of freedom are eliminated. Measuring everything around hover is done by attaching a large weight to a string and then attaching it to the X-Pro arm. The string is stretched just far enough to ensure the X-Pro cannot rise above hover without also elevating the weight. The fixation is done using strings attached to the X-Pro preventing it from rotating as well as making any roll movements. The measurements are then conducted by placing the weight on an electronic scale and then increasing the rotational speed of the rotor gradually. Then the lift provided by the rotor can be directly derived from the readings on the scale knowing that

$$F = m \cdot g \quad [\text{N}] \quad (5.15)$$

The setup is sketched in figure 5.13



**Figure 5.13:** The testsetup for measuring lift.

Controlling the rotational speed is done using the R/C transmitter and then measuring the actual number of revolutions using the hall sensor described in appendix B. This gives a limited number of steps due to the sticks resolution. However, this will provide the needed information for the relationship between rotational movement and the lift created.

As the only available measurement option on the scale is manual reading of the weight, the measurement is done when the system has reached steady state for that rotational speed, which means the dynamics of the rotor is not included in the measurements. With the X-Pro being in a fixed position with very limited pitch movement, the body hardly moves which means the body dynamics can be disregarded.

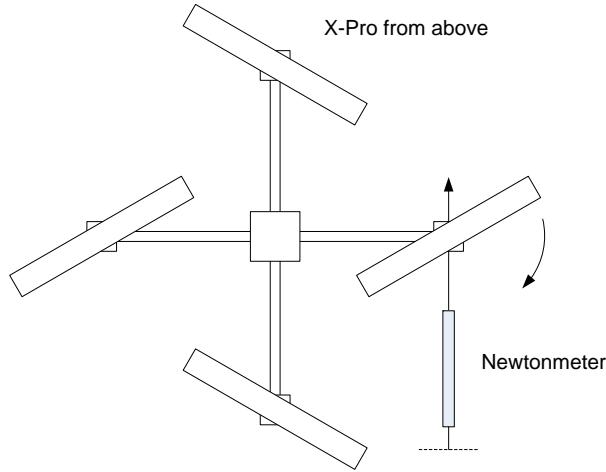
If the air inflow and rotor bandwidth is much faster than the bandwidth on the body, the effect of the loss of dynamics in the rotor model should be minimal.

Even with the X-Pro in a close to fixed position, the gyro alternates the rotational speed of the rotor during the measurements. This results in the weight displayed on the scale and the speed of the rotor not having a fixed value at any given controller stick setting. Hence the values gathered are affected by human errors and latency in the controller system, as the values of weight and rotational speed needs to be read at the same time.

### 5.3.2 Measuring drag

While measuring drag, pitch and roll of the X-Pro needs to be fixed. This is done by mounting the X-Pro on a frame without the ability to tilt. This way the newton meter can be fixed horizontally throughout the measurements, making sure that the only force acting on the newton meter is drag. The newton meter is also fixed perpendicular to the arm holding the rotor and as such parallel to the drag force created. As the X-Pro rotates due to the drag created by the rotation of the rotor the newton meter is stretched and is no longer parallel to the drag force, but considering the short distance and the inaccuracy of the readings, this is accepted. The test setup is sketched in figure 5.14

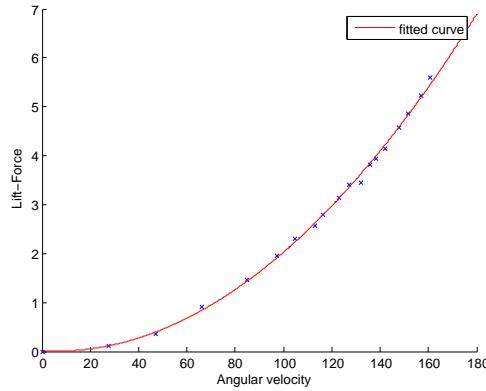
As for the lift measurements, the rotational speed is controlled using the manual controller and then measured using the Hall sensor. The force created is then read on the newton meter manually. The test setup using the fixed frame caused problems for the gyroscope which reacted with speed alterations, resulting in inaccuracies in the readings.



**Figure 5.14:** The test setup for measuring drag.

### 5.3.3 Parameter Estimation

Using the `fit` command in Matlab provides a fitting of the points to a given curve. The curve used is a second order polynomial. The following figures show the measurement data as well as the curve fitted to that data. The m-file including the measurement data and the curve fitting can be found on the CD. The resulting equations for the lift and for the drag are

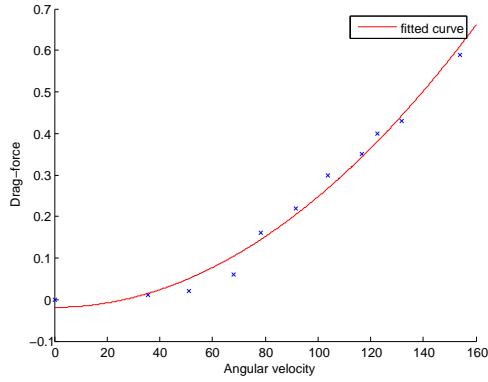


**Figure 5.15:** The measured lift and the curve fitted.

$$F_{Lift}(\Omega) = 0.2263 \cdot 10^{-3} \cdot \Omega^2 - 2.488 \cdot 10^{-3} \cdot \Omega + 11.97 \cdot 10^{-3} \quad [N] \quad (5.16)$$

$$F_{Drag}(\Omega) = 26.57 \cdot 10^{-6} \cdot \Omega^2 - 1.933 \cdot 10^{-6} \cdot \Omega + 18.84 \cdot 10^{-3} \quad [N] \quad (5.17)$$

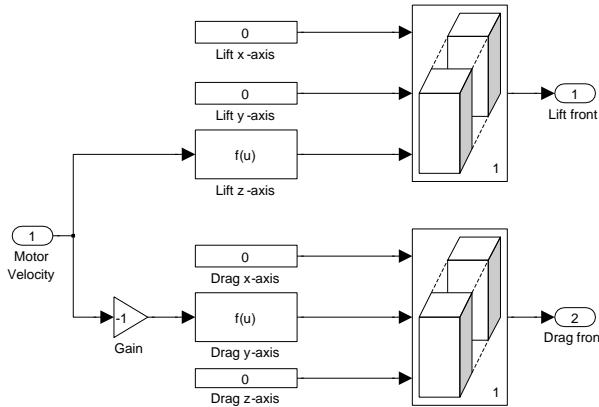
The measured data match a second order function acceptable, however lift is a better fit than drag. This could be caused by the test setup for drag turning out to be fairly unstable, thus resulting in the gyroscope altering the rotational speed.



**Figure 5.16:** The measured drag and the curve fitted.

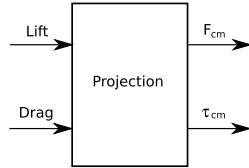
#### 5.3.4 Simulink Model

Since the dynamics of the rotor is included in the motormodel by keeping the rotor mounted on the motor during motormeasurements, the second-order polynomials can be used directly as models calculating the lift and drag based on rotational velocity. Lift generates movement on the z-axis while drag generates movement in either the x- or the y-axis and negative or positive, depending on which rotor the drag is calculated for. This is shown for the front rotor in figure 5.17.



**Figure 5.17:** The Simulink model used for the front rotor.

# Projection



*Figure 6.1: Model for the projection.*

## 6.1 Force

The force  $\vec{F}$ , illustrated in figure 6.1, is the sum of all forces and the force generated by the rotors will be positive and the force generated by the gravity affecting the X-Pro will be negative. The individual forces acting on the X-Pro has each been derived in chapter 5.3. In figure 4.2 the X-Pro is seen from behind. The forces that are produced by the rotors, are in the same direction as the coordinate system of the base in the z-axis. The forces from the rotors will be calculated as positive, and the gravity will be calculated as negative.

The X-Pro is affected by external forces while flying. To get a better understanding of the forces a short description is made:

### Gravity

Gravity is the only force affecting the X-Pro in the negative direction seen from the base coordinate system. The force attacks in the center of mass of the X-Pro, and will always point toward the ground. The force direction is always fixed to the orientations of the universal frame I.

## Rotor Lift

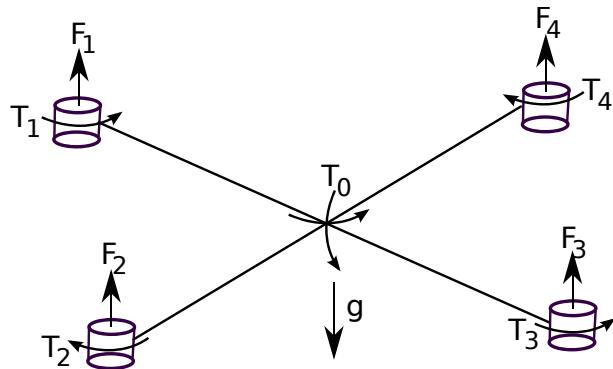
This force should essentially make the X-Pro fly by exceeding the gravity force. The lift attacks the X-Pro at the rotor hubs, and their position can each be expressed in coordinates of the X-Pro's frame B. It should be noted that the rotor lift is affected by the external environment, for instance the ground effect which occur when flying close to the ground. This effect results in increasing pressure below the blades and thereby giving a larger lift at the same rotational velocity of the rotors.

## Normal force

The normal force occurs if the X-Pro gets in contact with the ground. As the goal with the project is having the X-Pro stabilized in hover, the X-Pro should never touch the ground. Hence this force will be disregarded in this project

## Air resistance

Air resistance also affects the X-Pro. This is dependent on the surface of the X-Pro relative to the direction of the wind and wind speed. But as the X-Pro is going to fly indoor, these forces are relatively small. Furthermore it is decided that the X-Pro is only hovering and therefore will this force not be included in the model.



*Figure 6.2: Sketch of the forces and the torques acting on the X-Pro.*

The forces which will be included in the model are the gravitational force and the total rotor lift force.

$$\begin{aligned}
 {}^B\vec{F}_{tot} &= \sum_{j=1}^4 \left[ {}^B\vec{f}_{Lift,j} \right] - {}^B\vec{f}_g \\
 &= \sum_{j=1}^4 \left[ {}^B\vec{f}_{Lift,j} \right] - m_{tot} {}^B_R {}^I \vec{g}
 \end{aligned} \tag{6.1}$$

In equation (6.1),  ${}^I\vec{g}$  is the gravity vector in the universal frame I, the gravity only effects in the z axes and can be expressed as equation (6.2). The  ${}^B\vec{f}_{Lift,j}$  is the lift force from rotor  $j$ , found in section 5.3.

$${}^I\vec{g} = \begin{bmatrix} 0 \\ 0 \\ -9.82 \end{bmatrix} \quad [\text{m/s}^2] \quad (6.2)$$

## 6.2 Torque

In this section the torque will be described as a result of the drag and lift. The  ${}^B\vec{f}_{Lift,j}$  is a vector, with the lift force from the  $j$ 'th rotor.

$${}^B\vec{f}_{Lift,j} = \begin{bmatrix} 0 \\ 0 \\ f_j \end{bmatrix} \quad (6.3)$$

When a rotor generates a force, it leads to a torque on the X-Pro relative to the distance of the rotor from the center. The lift torque from one rotor is the cross product of the position and the lift force, which is the distance from the center of the base to the rotor. The  $\vec{P}_j$  vector represent the distance from the base to the rotor, in the x,y and z directions. The total lift torque,  $\tau_{Lift}$ , from the rotors can be calculated as the sum of the 4 lift torques.

$$\vec{\tau}_{Lift} = \sum_{j=1}^4 \vec{P}_j \times {}^B\vec{f}_{Lift,j} \quad (6.4)$$

When a rotor generates a lift force, it also generates a drag force. The drag is found in section 5.3. The torque from one rotor is the cross product of the drag force and the position. The total drag torque is the sum of the 4 drag torques.

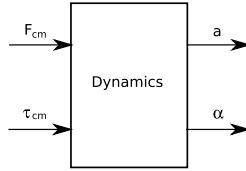
$$\vec{\tau}_{Drag} = \sum_{j=1}^4 \vec{P}_j \times {}^B\vec{f}_{Drag,j} \quad (6.5)$$

The total torque on the base, is the sum of  $\tau_{F_{Drag}}$  and  $\tau_{Lift}$

$$\vec{\tau}_{Total} = \vec{\tau}_{f_{Drag}} + \vec{\tau}_{Lift} \quad (6.6)$$

In this chapter the lift forces and drag forces from each of the 4 rotors has been projected and summed into total torques. The torques will be used in the next chapter to calculate the acceleration concerning the X-Pro.

# Dynamics



**Figure 7.1:** Model for the dynamics.

In this chapter the force and torque found in last chapter has to be calculated into a linear acceleration and an angular acceleration of the body frame, illustrated in figure 7.1. The two accelerations are going to be derived in the following two sections.

## 7.1 Forces to Accelerations

From Newton's second law its known that the following are valid in the universal frame [Cra89, p. 27]:

$${}^I \vec{F} = {}_B^I R {}^B F = m {}^I \vec{a} \quad (7.1)$$

where  ${}_B^I R$  describes the base frame relative to the universal frame on the X-Pro.

$${}^I \vec{a} = {}_B^I R {}^B \vec{F} \frac{1}{m} \quad (7.2)$$

$${}^I \vec{a} = {}^I \dot{\vec{v}} \quad (7.3)$$

The velocity of the body with respect to the universal frame can be expressed as a rotation of the velocity of the body, as written in equation (7.4).

$${}^I\vec{v} = {}^I_B R {}^B\vec{v} \quad (7.4)$$

$${}^I\dot{\vec{v}} = {}^I_B R {}^B\dot{\vec{v}} + {}^I_B \dot{R} {}^B\vec{v} \quad (7.5)$$

The derivative of the velocity, is found by using the chain rule on equation (7.5). This can further be rearranged to be an expression of the acceleration in the body frame.

$${}^B\dot{\vec{v}} = {}^B_I R {}^I\dot{\vec{v}} - {}^B_I R {}^I_B \dot{R} {}^B\vec{v} \quad (7.6)$$

$${}^B\dot{\vec{v}} = {}^B_I R {}^I\dot{\vec{v}} - {}^I_B R^T {}^I_B \dot{R} {}^B\vec{v} \quad (7.7)$$

The expression  ${}^I_B R^T {}^I_B \dot{R} {}^B v$  can be rewritten into  ${}^B\vec{\omega} \times {}^B\vec{v}$  [Bak02, p. 8] as the frame I is fixed, where  ${}^B\vec{\omega}$  is the angular velocity of the body. Equation (7.2) for the acceleration  ${}^I\dot{v}$  can be inserted instead, which results in the following equation.

$${}^B\dot{\vec{v}} = {}^B_I R {}^I_B R {}^B \vec{F} \frac{1}{m} - {}^B\vec{\omega} \times {}^B\vec{v} \quad (7.8)$$

The rotation  ${}^B_I R {}^I_B R$  can be reduced which results in the final equation

$${}^B\dot{\vec{v}} = {}^B \vec{F} \frac{1}{m} - {}^B\vec{\omega} \times {}^B\vec{v} \quad (7.9)$$

## 7.2 Moment to Angular Acceleration

The X-Pro is described as a rigid body, which is rotating with an angular velocity,  $\vec{\omega}$ , and has angular acceleration,  $\dot{\vec{\omega}}$ . The moment  $\tau$ , which causes this motions, is given by Euler's equation [Cra89, p. 196]:

$${}^B\tau = {}^I {}^B\dot{\vec{\omega}} + {}^B\vec{\omega} \times {}^I {}^B\vec{\omega} \quad (7.10)$$

where  $I$  is the inertia tensor of the body, whose origin is located at the center of mass. The inertia tensor can be written as the following [Cra89, p. 191]

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (7.11)$$

The description of how the Inertia tensor is found can be seen in appendix F. Equation (7.10) can be rewritten into an expression of the angular acceleration of the body.

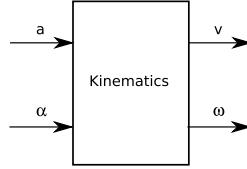
$${}^B\dot{\vec{\omega}} = I^{-1} {}^B\tau - I^{-1} ({}^B\vec{\omega} \times I {}^B\vec{\omega}) \quad (7.12)$$

In this chapter the linear acceleration and angular acceleration has been derived. The parameters will then be used in the kinematics to derived the final states, position, Euler angles and Euler rates.



# Chapter 8

## Kinematics



**Figure 8.1:** Model for the kinematics.

In this chapter the expressions found in the previously chapters will be used to derive the correlation between acceleration and the resulting velocity of the rigid body, illustrated in figure 8.1.

### 8.1 Linear Acceleration

The linear acceleration from the base coordinate system from section 7.9 yields

$${}^B \ddot{\vec{v}} = {}^B \vec{F} \frac{1}{m} - {}^B \vec{\omega} \times {}^B \vec{v} \quad (8.1)$$

$$(8.2)$$

The position is derived in 3 steps. The first step is to derive the linear velocity from the linear acceleration.

$${}^B \vec{v} = \int {}^B \dot{\vec{v}} \quad (8.3)$$

The next step is to derived the linear velocity of the base relative to the universal frame.

$${}^I \vec{v}_B = {}^I_R {}^B \vec{v} \quad (8.4)$$

In the last step the position of the base frame will be described relative to the universal frame.

$${}^I P = \int {}^I \vec{v}_B \quad (8.5)$$

## 8.2 Angular Velocity

The angular acceleration of frame B is used to find the angular velocity in the universal frame I.

$${}^B\dot{\vec{\omega}} = {}^I\tau^{-1} - {}^I\tau^{-1}({}^B\vec{\omega} \times {}^I{}^B\vec{\omega}) \quad (8.6)$$

$${}^B\vec{\omega} = \int {}^B\dot{\vec{\omega}} \quad (8.7)$$

The velocity of the frame B and the rotation matrix can be used to find the Euler angles  $\theta$  and the Euler rates,  $\dot{\theta}$ . It can be derived from the kinematic differential equation for the (3-2-1) sequence [Bak02, p. 25].

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{\cos(\theta_2)} \begin{bmatrix} \cos(\theta_2) & \sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) \sin(\theta_2) \\ 0 & \cos(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \cos(\theta_2) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (8.8)$$

$$\vec{\theta} = \int \dot{\vec{\theta}} \quad (8.10)$$

It is not possible directly to calculate  $\theta$  and  $\dot{\theta}$  as they depend on each other, so they will be found in an iterative process.

The rotation matrix  ${}^B_R$  can be found by using the Euler (3-2-1) rotation, when the angles,  $\theta$ , are known.

$$C_3(\theta_3) = \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.11)$$

$$C_2(\theta_2) = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ 0 & 1 & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \quad (8.12)$$

$$C_1(\theta_1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) \\ 0 & -\sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \quad (8.13)$$

The final rotation matrix yields:

$${}^I_B R = C_1(\theta_1) C_2(\theta_2) C_3(\theta_3) =$$

(8.14)

$$\begin{bmatrix} \cos(\theta_2)\cos(\theta_3) & \cos(\theta_2)\sin(\theta_3) & -\sin(\theta_2) \\ \cos(\theta_3)\sin(\theta_1)\sin(\theta_2) - \cos(\theta_1)\sin(\theta_3) & \sin(\theta_1)\sin(\theta_2)\sin(\theta_3) + \cos(\theta_1)\cos(\theta_3) & \sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\sin(\theta_3) + \cos(\theta_1)\sin(\theta_2)\cos(\theta_3) & \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_3) & \cos(\theta_1)\cos(\theta_2) \end{bmatrix} \quad (8.15)$$

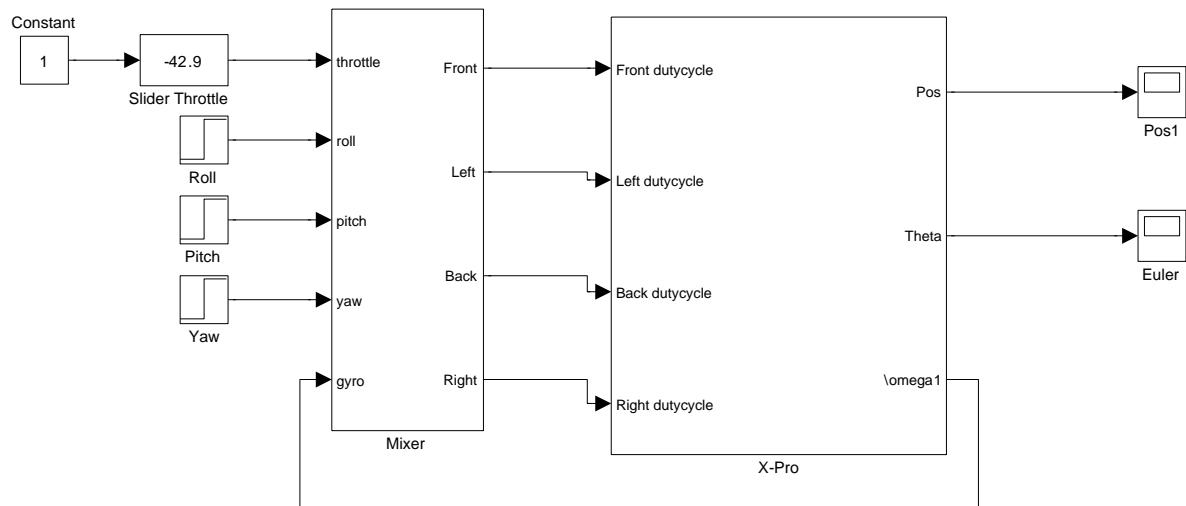
The desired states that describes the X-Pro has now been derived. In the next chapter the equation of the states will be put together to form the final model. The model will be implemented and tested in Simulink.



# Chapter 9

## Combining the Model

The model needed for designing a controller for the X-Pro, is derived in the last 4 chapters. The model needs to reflect the reality, using input values in the range  $[100; -100]$  and an output vector including the position and the Euler angles  $[P; \theta]$ . The model will be implemented and simulated in Simulink. In figure 9.1 the model of the X-Pro is shown, not including the mixer that receives the input signal in the range  $[100; -100]$ . The mixer is described and verified in appendix A.



**Figure 9.1:** The subsystems in the X-Pro model.

### Motor

The motor blocks take duty cycle as input and provides angular velocity,  $\Omega$ , as output, the subsystems are single input single output transfer functions, derived in the section 5.2 in chapter 5.

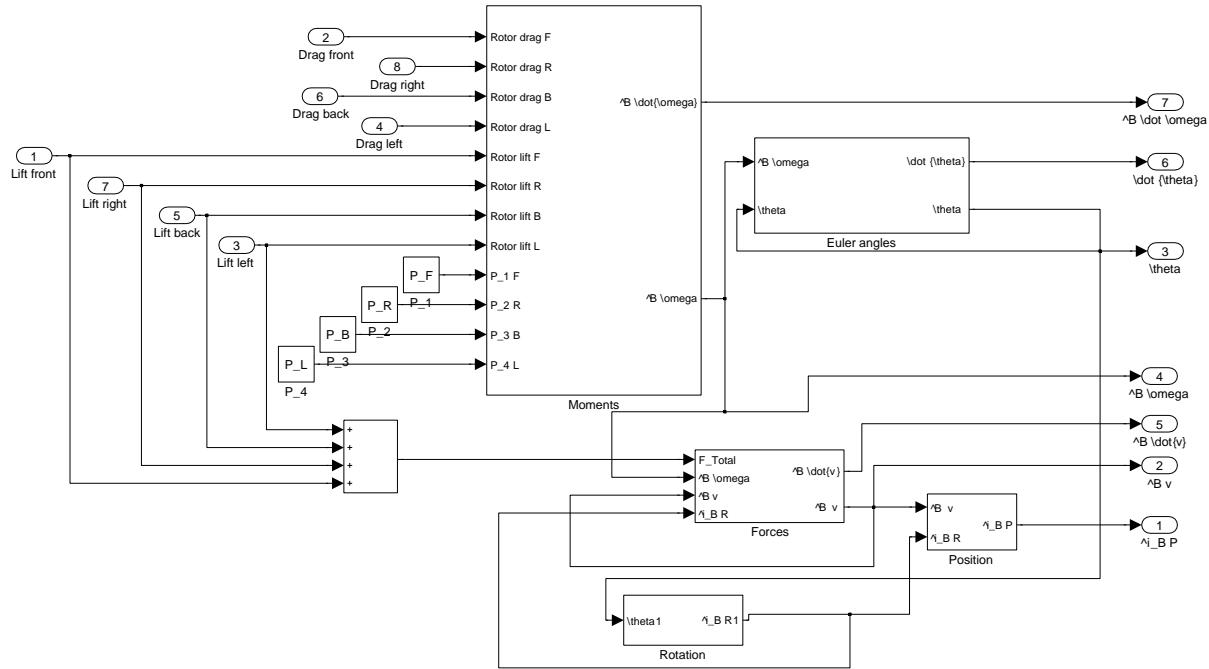
## Rotor

The actual force each rotor produces is calculated in the next subsystem, the two polynomial's which are used to calculate the force and drag takes the angular velocity as input. The two polynomials are derived in section 5.3 rotor in chapter 5. The forces the rotors produce are used in the body subsystem.

## Body

The body subsystem takes the forces from the individual rotors and the constant distance from the rotor to the base as inputs. The front-, right-, back- and left rotor are expressed as  $P_F$ ,  $P_R$ ,  $P_B$  and  $P_L$  respectively and are measured in meters. The body subsystem structure is illustrated in figure 9.2.

$$P_F = \begin{bmatrix} 0,455 \\ 0 \\ 0,073 \end{bmatrix} \quad P_R = \begin{bmatrix} 0 \\ -0,455 \\ 0,073 \end{bmatrix} \quad P_B = \begin{bmatrix} -0,455 \\ 0 \\ 0,073 \end{bmatrix} \quad P_L = \begin{bmatrix} 0 \\ 0,455 \\ 0,073 \end{bmatrix} \quad (9.1)$$



**Figure 9.2:** The body subsystem in the X-Pro model.

## Moments

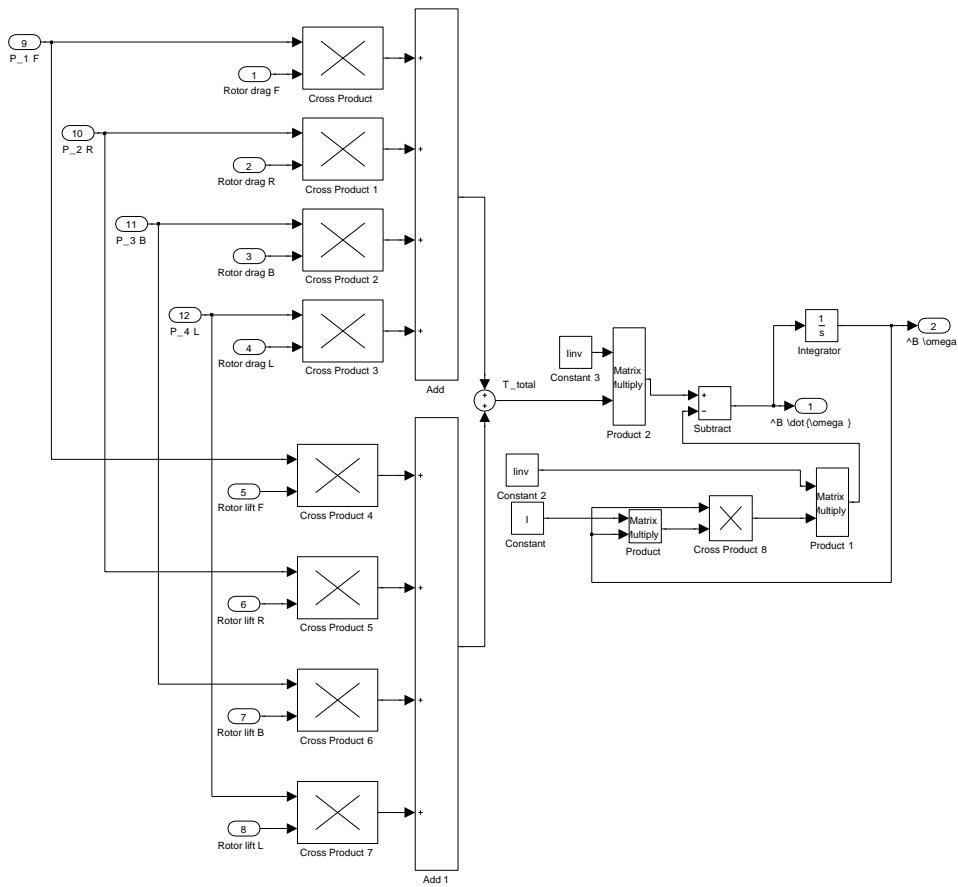
The moments and torques the X-Pro generates, is calculated in the moment subsystem. The moments subsystem use the distances and the forces, to calculate the total torque in  $[x, y, z]$

directions. The torque is used to find,  ${}^B\dot{\omega}$  and  ${}^B\ddot{\omega}$  from equation 9.2

$${}^B\dot{\omega} = I^{-1} {}^B\tau - I^{-1} ({}^B\vec{\omega} \times I {}^B\vec{\omega}) \quad (9.2)$$

$${}^B\ddot{\omega} = \int {}^B\dot{\omega} \quad (9.3)$$

The  ${}^B\omega$  and  ${}^B\dot{\omega}$  is the output of the subsystem.



**Figure 9.3:** The moments subsystem in the X-Pro model.

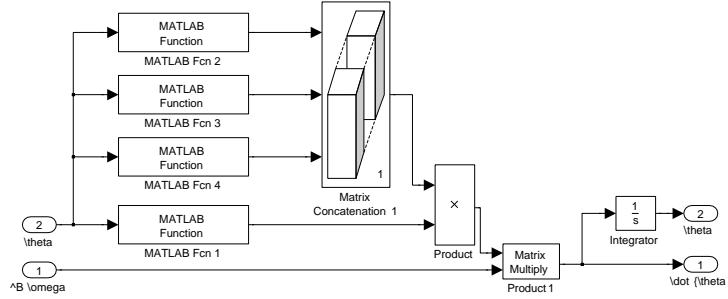
The angular velocity in frame B is calculated in the subsystem velocity show in figure 9.3. The velocity subsystem returns the  ${}^B\omega$  and  ${}^B\dot{\omega}$ .

The inertia matrix is calculated in appendix F and yields:

$$I = \begin{bmatrix} 0,15 & 0 & 0 \\ 0 & 0,15 & 0 \\ 0 & 0 & 0,28 \end{bmatrix} \quad [\text{m}^2\text{kg}] \quad (9.4)$$

## Euler Angles

The Euler angles and rates are calculated in the subsystem Euler angles seen in figure 9.4.



**Figure 9.4:** The Euler angles subsystem in the X-Pro model.

The Euler angles  $\theta$  and  $\dot{\theta}$ , is calculated in the subsystem Euler angles. The subsystem calculates the Euler rates,  $\dot{\theta}$  from

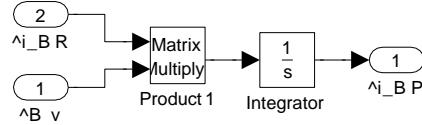
$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{\cos(\theta_2)} \begin{bmatrix} \cos(\theta_2) & \sin(\theta_1) \sin(\theta_2) & \cos(\theta_1) \sin(\theta_2) \\ 0 & \cos(\theta_1) \cos(\theta_2) & -\sin(\theta_1) \cos(\theta_2) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (9.5)$$

When the Euler rates are calculated,  $\theta$  can be found by

$$\vec{\theta} = \int \dot{\theta} \quad (9.6)$$

## Position

The position of the X-Pro model, in [x, y, z], is calculated in the subsystem position.



**Figure 9.5:** The position subsystem in the X-Pro model.

The position calculation in figure 9.5 is found by

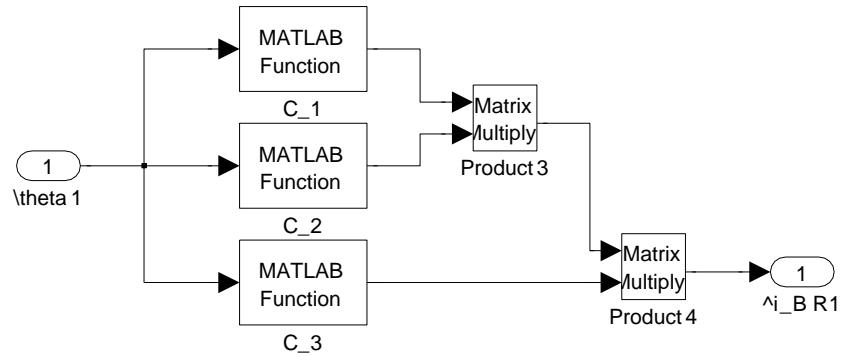
$${}^B \vec{v} = \int {}^B \dot{\vec{v}} \quad (9.7)$$

$${}^I \vec{v}_B = {}^I_B R {}^B \vec{v} \quad (9.8)$$

$${}^I P = \int {}^I \vec{v}_B \quad (9.9)$$

## Rotation

The rotation matrix is calculated from the 3-2-1 Euler rotation. Using the  $\theta$  vector found in the angles subsystem.

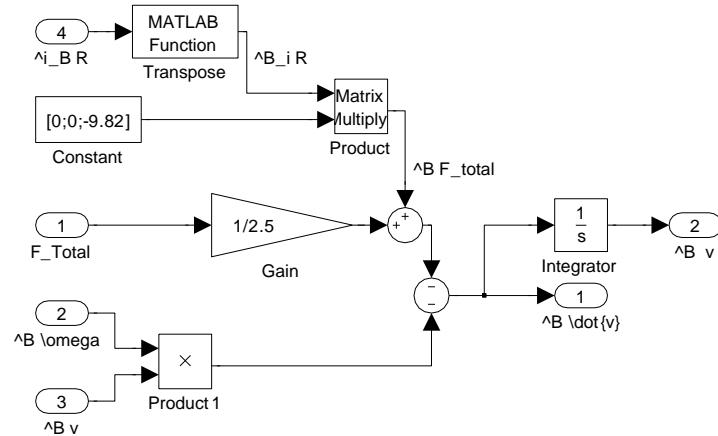


**Figure 9.6:** The rotation subsystem in the X-Pro model.

The rotation matrix in figure 9.6 is found by applying the result from section 8.

## Forces

The velocity of the X-Pro is calculated in the subsystem forces shown in figure 9.7.



**Figure 9.7:** The forces subsystem in the X-Pro model.

The equations for the velocity and the derivative yields

$${}^B\dot{\vec{v}} = {}^B\vec{F} \frac{1}{m} - {}^B\vec{\omega} \times {}^B\vec{v} \quad (9.10)$$

$${}^B\vec{v} = \int {}^B\dot{\vec{v}} \quad (9.11)$$

The X-Pro model provides the mapping from the output of the R/C transmitter to the output vector  $[P; \theta]$ . To verify the model, a data recording of input and output during flight is gathered.

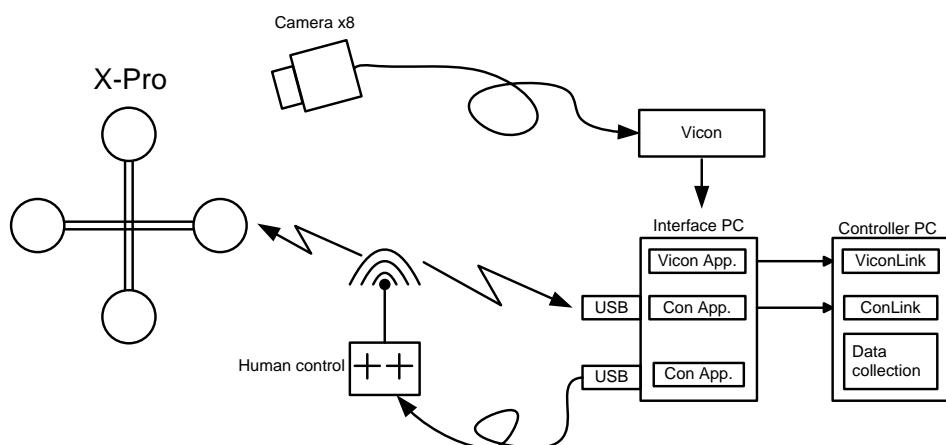
# Chapter 10

## Non-Linear Model Verification

A complete mathematical model was derived in chapter 4 to 8, and needs to be verified. The full mathematical model is implemented in Simulink. To verify the Simulink model it has to be compared to the X-Pro, this requires data from the X-Pro flying in hover. Then the measured control input can be sent through the model, and the output angles can be compared to the measured angles from the X-Pro.

### 10.1 Obtaining the test Data

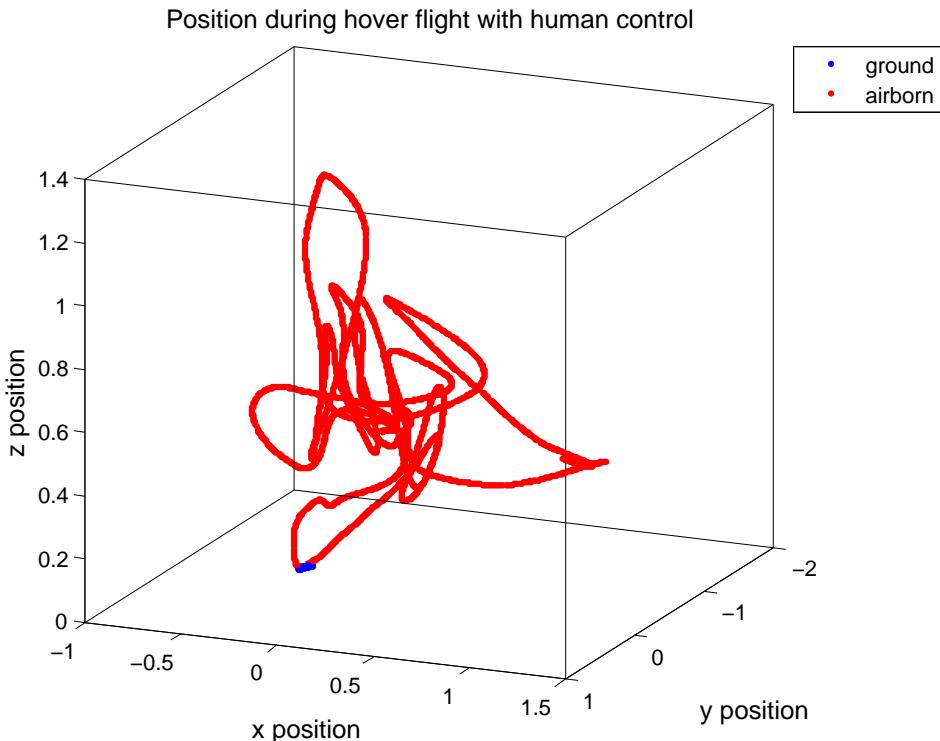
The data needed to verify the mathematical model is the output from the R/C transmitter and the Vicon data  $[P; \theta]$ . The data sent from the R/C transmitter needs to be mapped, into what is received on the Controller PC. The mappings and a further description of the signal flow, can be found in the analysis in section 2.5. The data will be measured and saved in Matlab. The test setup can be seen in figure 10.1.



**Figure 10.1:** Test setup for data collection.

The X-Pro has to be controlled by a person, that tries to stabilise the X-Pro in hover. During the period where it is in hover, the person controlling the X-Pro tries to do some step inputs on yaw, pitch and roll. These step inputs can then be used to verify, whether the Simulink model are responding as the X-Pro. From the test of the X-Pro it was experienced that pitch and roll control inputs, were difficult to do step inputs on. This can be caused of the room which is small. To avoid from crashing the X-Pro into the wall the step inputs are only done for yaw. To verify pitch and roll normal control data are going to be used.

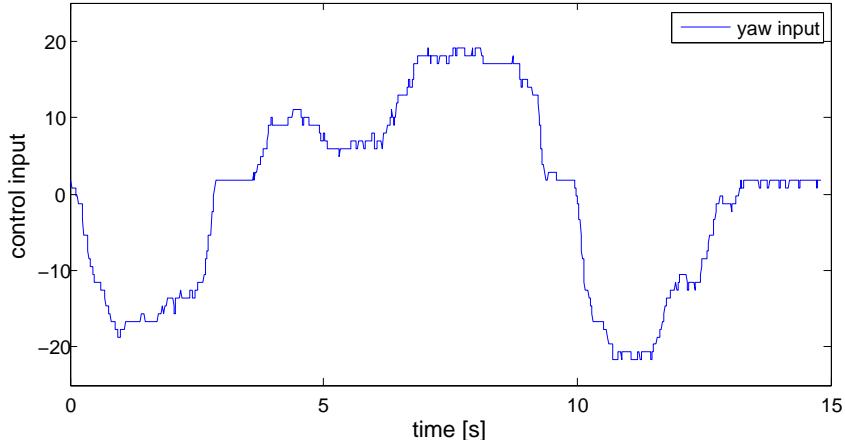
A 3D plot of one of the test with manual control of the X-Pro, trying to keep the X-pro in hover, is shown in figure 10.2. As it can be seen the person controlling the X-Pro had problems just keeping it in hover.



**Figure 10.2:** 3D plot of manual control attempting to keep the X-Pro in hover.

## 10.2 Collected Data

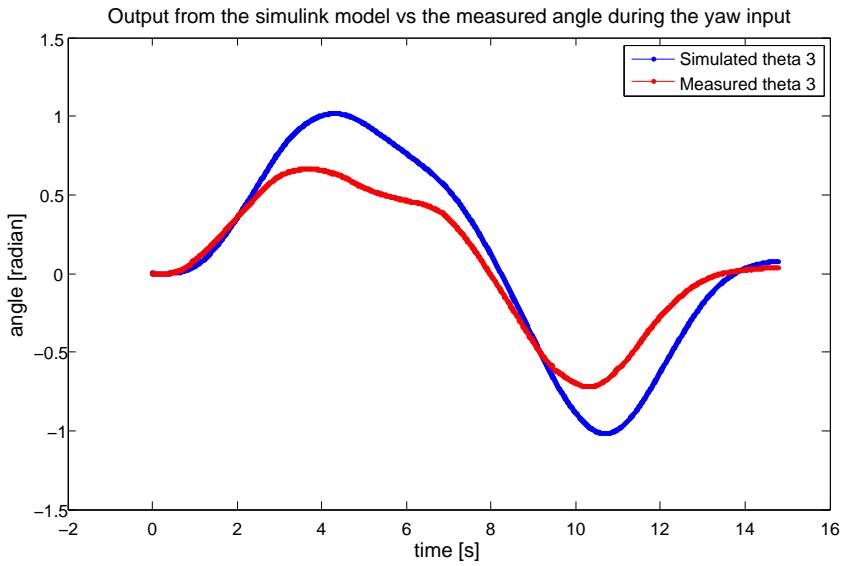
There were performed several tests. From these test the best input/output relations were picked out. For the yaw test the best period where only the yaw input were changed and only the yaw angle were affected and for pitch and roll there were no period where only one control signal dominated, so they were picked out together and tested together. The measured control data are mapped to the input for the model. The input for yaw are shown in figure 10.3. The measured angles during the flight are shown in figure 10.4, when they are compared to the simulated data. The offset for the control signals have been removed as they are not at part of the model. The control input can then be set as input for the Simulink model.



**Figure 10.3:** Collected control input data for yaw. The data have been mapped to the input.

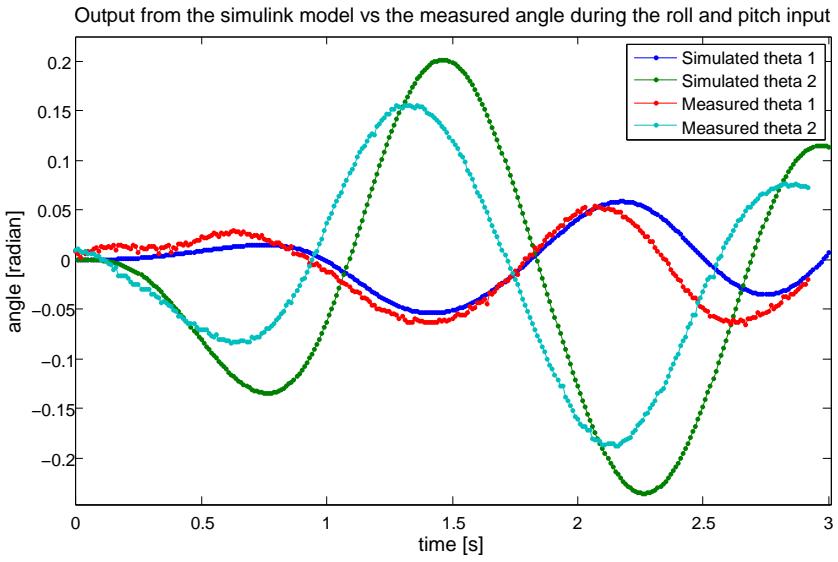
### 10.3 Model Data

The mapped control input are put through the Simulink model, and the  $\theta$  angle are recorded to use for verification against the measured angle. During yaw test, the input of pitch and roll are set to zero and the throttle are set to -41, which is the steady state. For pitch and roll test, yaw input were set to zero. The  $\theta$  output from the model during the yaw test compared to the measured angle are shown in figure 10.4, and the output from the pitch and roll test are shown in figure 10.5.



**Figure 10.4:** Angle from the Simulink model compared to the measured angle during the yaw test.

As it can be seen in figure 10.4 the Simulink model compared to the X-Pro fit acceptable. The model reacts more to the control input than the X-Pro did. This could be due to wrong values in the inertia tensor or the drag coefficient being to large.



**Figure 10.5:** Angle from the Simulink model compared to the measured angle during the pitch and roll test.

In figure 10.5 pitch and roll outputs are shown together with the measured angles. It can be observed that it moved in the right direction, but with a larger variation on the pitch.

In both cases there are also delay in the Simulink model, compared to the measured angles. This is probably due to a delay in the measurements from the test.

The tests show that the non-linear model reacts like the X-Pro would, when the same signal are sent through. It is therefore assumed that the model of the X-Pro is acceptable.

# Chapter 11

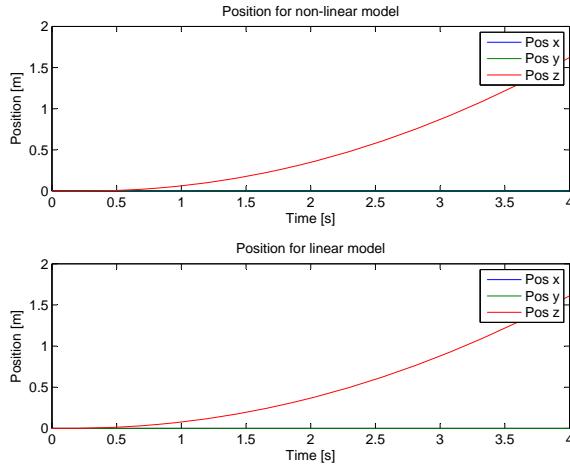
## Linear Model Verification

The linear model is derived in appendix C, and a state space representation has been established from the model. The representation will be used in order to develop a state space controller. First the linear model will be compared to the non-linear model. This is done by comparing the two outputs on both models, when adding a step to the inputs. The input steps will be throttle, yaw, pitch and roll.

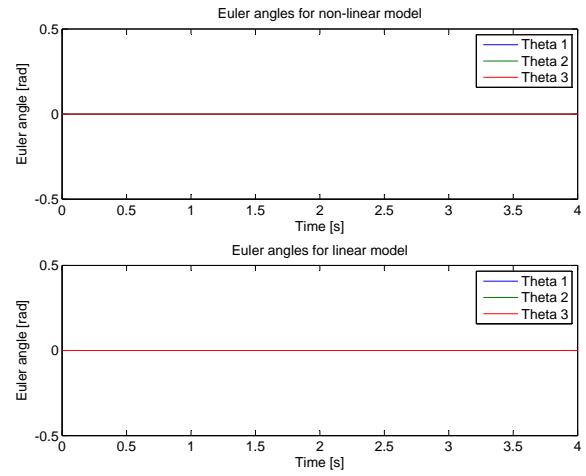
A certain amount of throttle must be applied in order to simulate hover in the nonlinear model. Without controller on the non-linear model, the plots will show an increasing change in z position, even when the throttle input are set to its equilibrium point. In order to compare the two models, a constant is added on the linear model, to get the models to act alike. All the steps are performed with step size 2 and only a step in one direction at the time.

### 11.1 Throttle

In figure 11.1 a step applied to the throttle on both the non-linear and the linear model, is shown. Both models react like expected since the z position is increasing. It is not expected that the throttle step should change the Euler angles, which can be verified in figure 11.2. The two models keeps the same angle values.



**Figure 11.1:** Position values when applying a step to the throttle input on the linear and non-linear model.

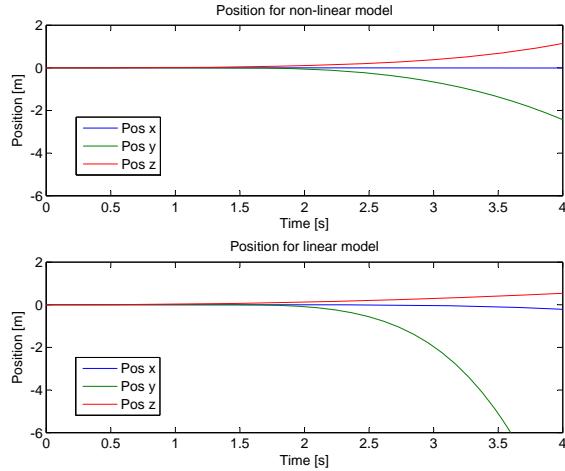


**Figure 11.2:** Euler angle values when applying a step to the throttle input on the linear and non-linear model.

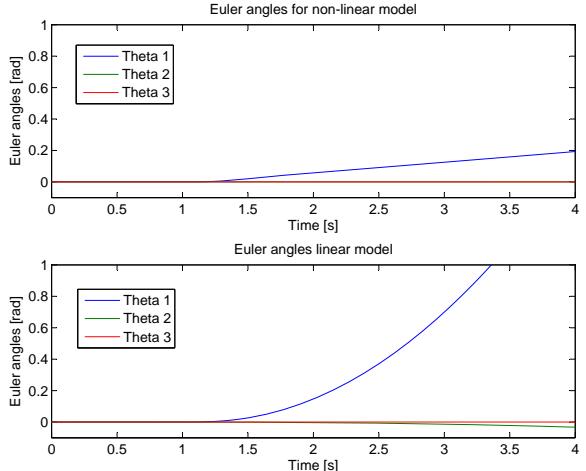
## 11.2 Roll

When applying a step on the roll input the models results in positive x direction values, shown in figure 11.3. The linear model is decreasing faster than the non-linear, which is caused by the gyroscope effect on the non-linear model.

The step on the roll input in figure 11.4 have a positive Euler angle effect around the x axis. In



**Figure 11.3:** Position values when applying a step to the roll input on the linear and non-linear model.



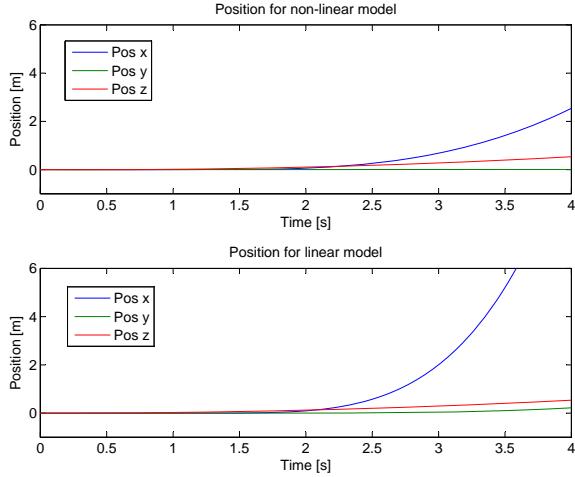
**Figure 11.4:** Euler angle values when applying a step to roll input on the linear and non-linear model.

the figures it can be seen that the models acts as expected.

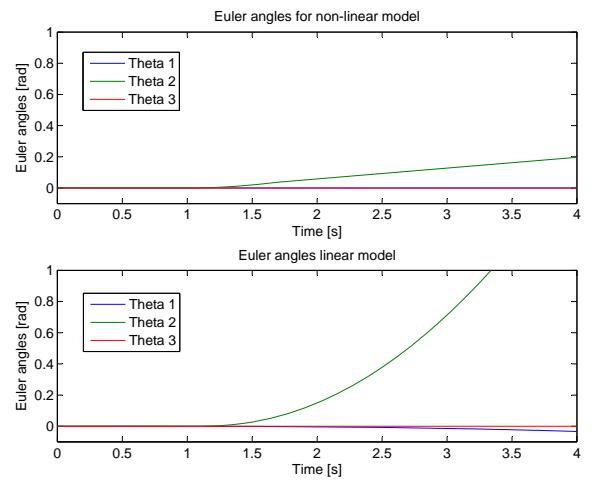
## 11.3 Pitch

The step applied on the pitch input is shown in figure 11.5 and makes both models move in the positive x direction. Both models reacts as expected, but as on roll step, the non-linear is less aggressive due to the gyroscope effect.

The angles shown in figure 11.6 for the roll step, results in positive y values.



**Figure 11.5:** Position values when adding a pitch input step on the linear and non-linear model



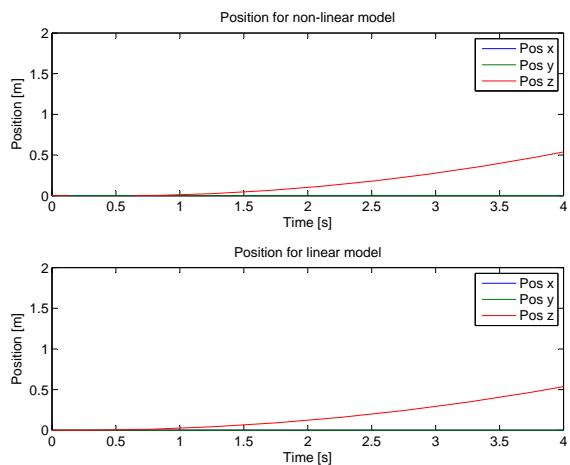
**Figure 11.6:** Euler angle values when adding a pitch input step on the linear and non-linear model

## 11.4 Yaw

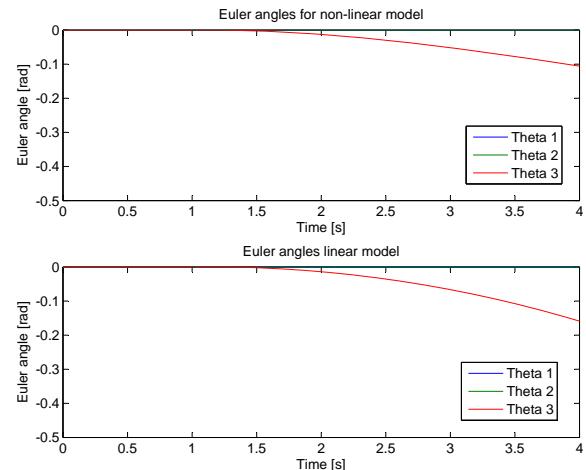
The step applied to the yaw input is shown in figure 11.7 and makes the models change in z values, which do not affect the x and y position of the models.

The positive step on the yaw input results in a negative Euler angle on the z axis.

The linear model react like the non-linear model, but is more aggressive. The linear model is sufficient enough, and will be used in designing the state-space controller.



**Figure 11.7:** Plot of the positions affected by a step on pitch input.



**Figure 11.8:** Plot of the Euler angle affected by a step on the yaw input.

# Chapter 12

## Model Discussion

In the modelling part, the X-Pro has first been examined concerning force generation. A model for the motors on the X-Pro has been derived using system identification. The motors has been model with gear and rotor attached which results in, dynamics and non-linear elements from the rotors will be included.

The forces and torques produced by the rotors has been projected to the X-Pro. The X-Pro has been modelled as a rigid body concerning the dynamics and kinematics. All the separate elements has been combined to form a complete non-linear model.

The non-linear model has been implemented in Simulink and compared to the real physically X-Pro and the model is validated. When doing the test flight, affection by the ground- and wind effect has been experienced.

The non-linear model has been linearised in appendix C and a linear model has also been implemented in Simulink. The linearised model has been compared to the non-linear model by applying steps to the inputs. It was experienced that the linear model was sufficient enough.



# Part III

## Controller



# Chapter 13

## PID Controller

The controller design can be seen as an iterative process. To get a feel for how the system works and how the X-Pro flies, a PID controller is designed for yaw, pitch, roll and throttle. Controlling the throttle in this case is controlling the altitude of the X-Pro and not a fixed throttle setting. The PID controller does not take the model into account and as such can only be seen as a preliminary controller. Afterwards a single state space controller is designed, controlling all degrees of freedom.

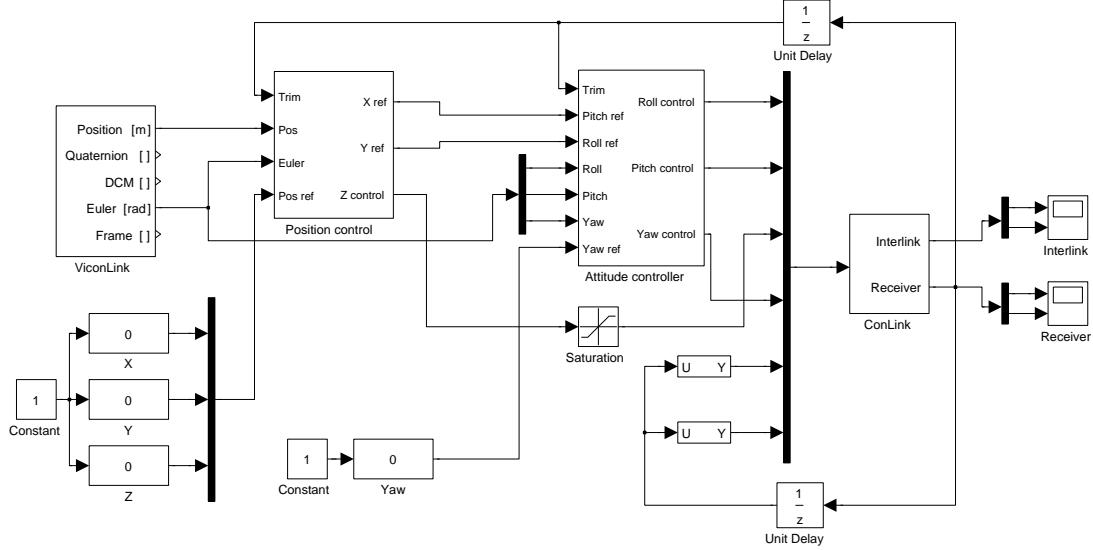
Controlling throttle, pitch, yaw and roll can be done independently with a PID controller on each. The stand that the X-Pro is mounted on gives the possibility to create attitude control without the risk of crashing the X-Pro due to bad position controller tuning. This way, control can be established on one degree of freedom at a time. The first two degrees that are controlled are pitch and roll, which for the X-Pro requires similar control, while completely neglecting yaw and position. When pitch and roll are working acceptable, yaw can be implemented to prevent the X-Pro from rotating.

After getting the attitude control to work acceptable, the X-Pro can be dismounted from the stand and placed on the ground. At this point the x and y position controller can be implemented, while keeping a human control on the throttle until the controller works for the x and y position. Lastly a controller for the throttle or z-axis movement can be implemented.

### 13.1 Implementing the PID Controller

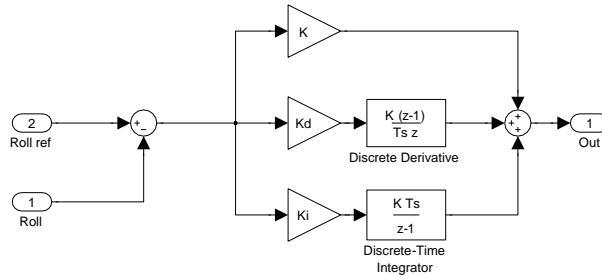
The Conlink block takes 6 control signals as inputs from which 4 are used for throttle, pitch, yaw and roll. The last two channels are not used for control of the X-Pro. The control signals are forwarded to the X-Pro through the ConLink block. The feedback signals are the angles and position measured by the Vicon Motion Tracking System. Both blocks are explained in appendix E. The setup to control the X-Pro is shown in figure 13.1. The setup consists of 2 controller blocks, one for the position and one for the attitude. The attitude controller block takes the output of the position controller as reference, except for yaw which does not have an outer position control loop. The model has slider bars to control the position and yaw reference

and a saturation on the throttle output to prevent too high control signals. Trim on the model is a feedback signal that registers when the controller is enabled and then resets all integral parts.



**Figure 13.1:** The Simulink setup for controlling the X-Pro.

The attitude controller consists of 3 PID controllers, one for each of pitch, yaw and roll. The PID controller for roll is shown in figure 13.2 and is done similar for pitch and yaw. To tune the controller the values of  $K$ ,  $K_d$  and  $K_i$  are changed. For all the attitude controllers it is chosen not to use any  $K_d$  as the X-Pro is shaking slightly when flying. Because of the precision in the Vicon Motion Tracking System, this is registered as slight fast movement by the X-Pro and a derivative part will provide high signal fluctuation.

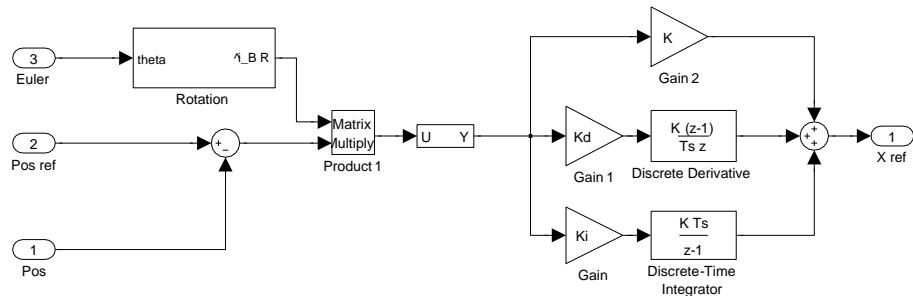


**Figure 13.2:** The Simulink block for controlling roll.

Since the measured angle input for pitch and roll are in radians and the output is a control value from -100 to 100, the gains needs to be rather large. Testing the controller on the X-Pro mounted on stand and fixated to remove movement on pitch and yaw provided the first values for roll. The proportional gain  $K$  is set to 100 and the differential and integral gain,  $K_d$  and  $K_i$ , are set to zero, resulting in just a proportional controller. This provides a satisfactory response on the attitude. A steady state error is acceptable on the angle, as any unwanted movement due to the attitude being off results in the position controller block to react, which means the attitude controller can work without an integral part. The same is valid for the pitch but the yaw control is different. It requires an integral part in order to counter the steady state error as there is no outer loop for yaw. The proportional gain  $K$  and the integral gain  $K_i$  are found to

be 40 and 5 respectively.

With the attitude controlled in place, a position controller is set up providing the reference point for the attitude control. Similar to the attitude controller, the position controller consists of 3 PID controllers, one for each of the x, y and z direction. The position control takes the measured position as input together with a provided reference point. The measured position is the distance from the center point defined in the Vicon system. As the coordinate system of the X-Pro is only aligned with the Vicon coordinate system once yaw is at a certain angle, the position needs to be transformed into the X-Pro coordinate system. This is done using the rotation matrix calculated in chapter 8 by providing the measured Euler angles as input as well. The position controllers take values in meters as input and the output is in radians. As a fairly small angle will provide a shift in movement, the gains in the position control needs to be rather small. Otherwise the different PID controller blocks for x, y and z is similar to those used in the attitude controllers. The Simulink block for control in the direction of the x-axis can be seen in figure 13.3.



**Figure 13.3:** The Simulink setup for controlling movement on the X-axis.

A change in the position requires a change in the velocity of the X-Pro which is the result of a change in the attitude. In order to counter the need of a velocity controller, the position controller requires a derivative part. The values for the constants  $K$ ,  $K_d$  and  $K_i$  in the position controller in the x and y direction are found to be 0.14, 0.08 and 0.06 respectively. The z axis provides a control signal that is to be sent to the X-Pro directly without going through another controller block, so the gains here needs to be larger. The values for  $K$ ,  $K_d$  and  $K_i$  in the magnitude of 100, 40 and 50 respectively provides an altitude control for the X-Pro.

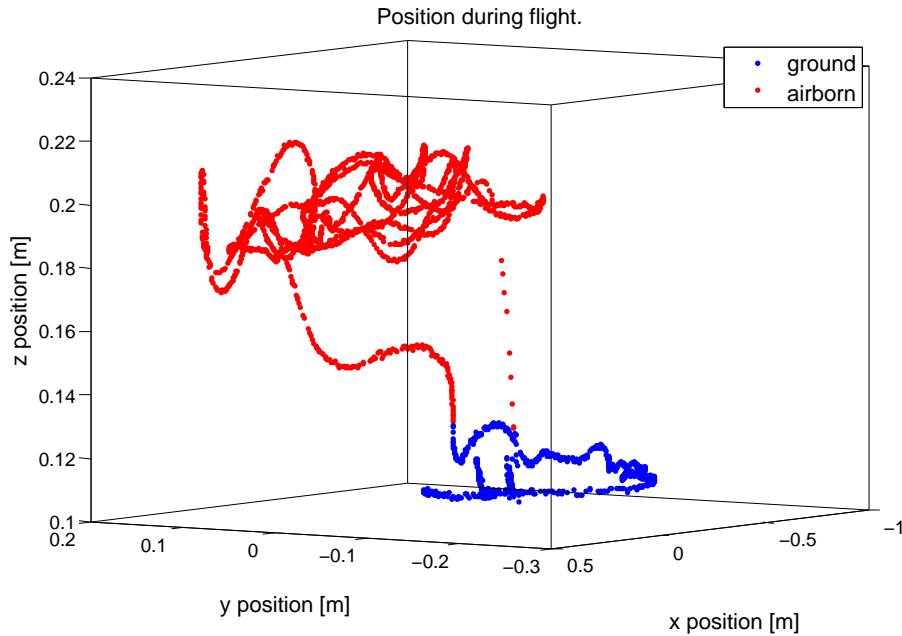
### 13.1.1 Testing the PID controller

The PID controller is tested according to the accept test which have been described in section 3.1 on page 14. During the test it was observed that the air flow in the room had a big influence on the performance. By opening the door to the room, altering the airflow, the performance was improved drastically. The two tests were performed with the door open. The results from the two test are outlined in the following.

#### Test 1: Keeping position and angle

The X-Pro was brought to hover and the angel and position were recorded. The complete flight are shown in a 3D plot in figure 13.4. The reference point for x and y is 0 meters and the altitude

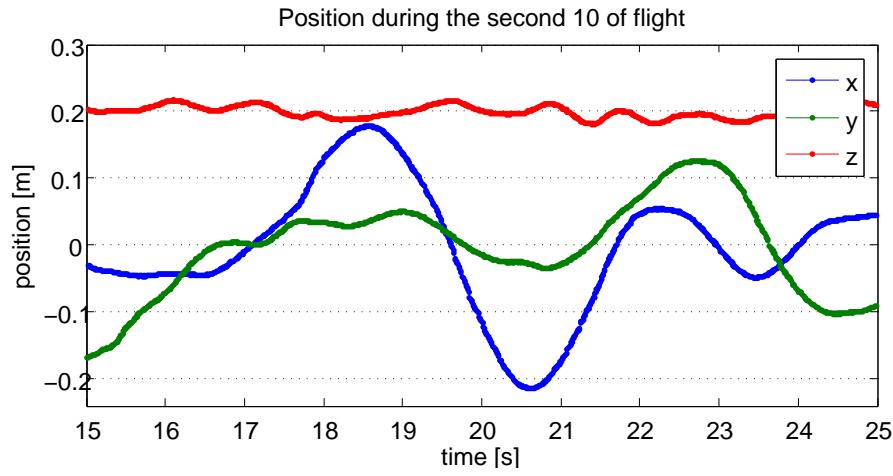
is set to 0.2 meters.



**Figure 13.4:** A 3D plot of the position of the X-Pro during the PID hover test.

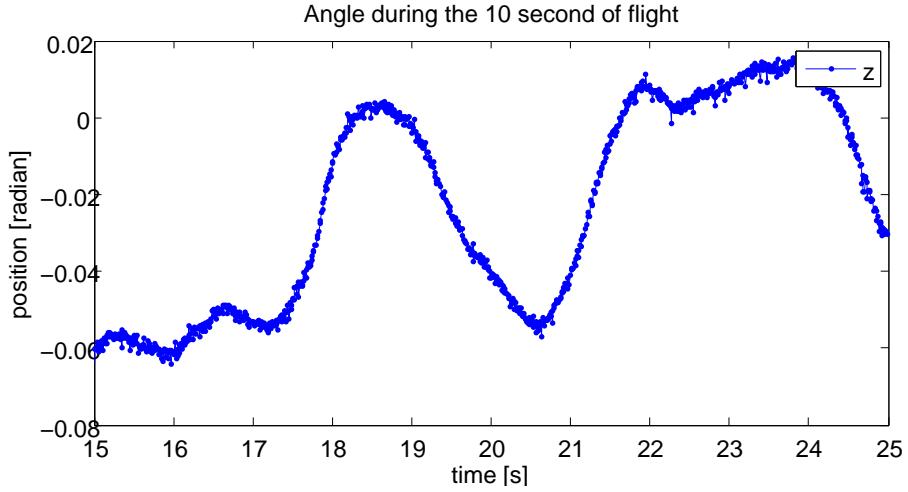
A video recording has been made of the flight, which can be seen on CD\video\PID perfect tuned.

10 seconds of the flight are chosen for analysis of the x and y position and the yaw angle. The position and angles for these 10 seconds are shown in figure 13.5 and 13.6 respectively.



**Figure 13.5:** The position of the X-Pro during the PID hover test.

In figure 13.5 it can be seen that the X-Pro's position fluctuate a little during the flight. The maximum deviation from the reference for the x and y position were found to be 0.22 and 0.17 meters. This is more than the  $\pm 0.1$  meter defined in the requirements. The altitude controller



**Figure 13.6:** The angle of the X-Pro during the PID hover test.

performed better as it differed with maximum 0.02 meter from the reference of 0.2 meter, which is within the required  $\pm 0.05$  meters. The X-Pro keeps the yaw angle with a maximum deviation of 0.064 radian which is within the required  $\pm 0.1$  radian.

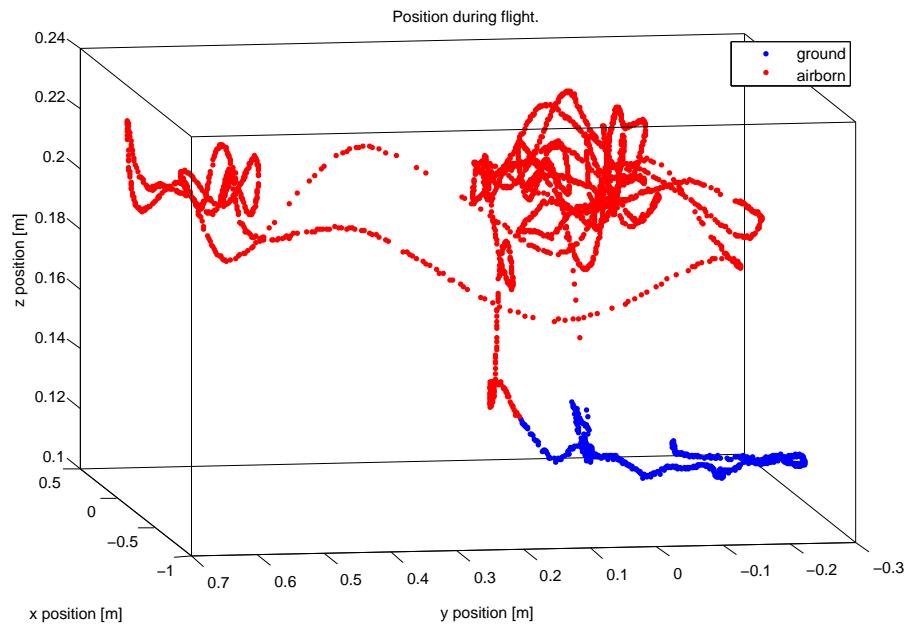
### Test 2: Changing position

The second test described in section 3.1 involves changing the reference position of the X-Pro, and monitoring how fast the X-Pro changes the position and settles at the new position. In this test the reference for the y position is changed from 0 to 0.5 meters and back to 0 again a while later. The whole flight was recorded and a 3D plot of the flight can be seen in figure 13.7

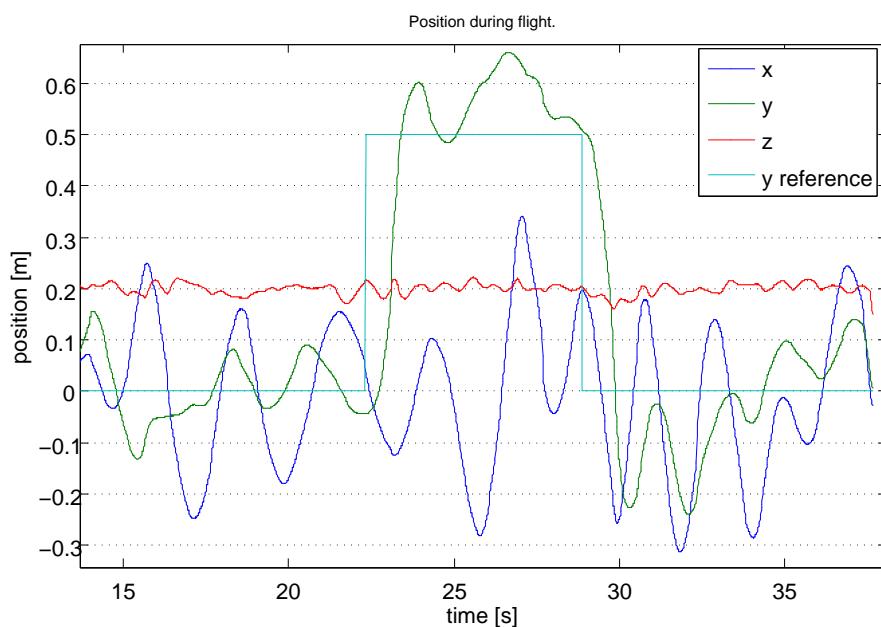
The measured positions of the X-Pro together with the reference signal for the y position is also shown in figure 13.8.

The graph in figure 13.8 shows that the X-Pro is fast to change the position. It overshoots with about 20%, but it is not much compared to the normal movement of the X-Pro. The rise time from the point where the step was given to it was at 90% of the reference was measured to be 1.01 second. At that point the X-Pro was fluctuating around the hover point as in the position test.

The combined model as shown in figure 13.1 results in the X-Pro trying to hover above the Vicon 0,0 coordinate, with the ability to move the fixpoint using the sliderbars. The controller does not keep it in the correct position all the time and not within the requirements outlined. It is concluded that when the X-Pro is exposed to movement with reference changes, the controller acts satisfactory and moves the X-Pro in place within the required 2 seconds. The controller does not keep the position steady as external forces affect the X-Pro, especially in the form of wind flow. This suggests that the controller is not fast enough to counter these changes immediately. It is assumed that the result can be improved by tuning the PID controllers more, but it is expected to achieve an overall better result with a model based controller.



**Figure 13.7:** A 3D plot of the position of the X-Pro during the PID move test.



**Figure 13.8:** Position of the X-Pro during the PID move test.

# Chapter 14

## State Space Controller

The X-Pro is a multiple input - multiple output system. Instead of using a PID controller as in the last chapter, a controller based on the states of the system is used. As a linearised controller is wanted, all equations that describes the system must be linearised. This is done in appendix C on page 105 and the representation will be used in this chapter.

The state space representation of the system is a connection of subsystems. As the subsystems, body/gear, rotor, motor and mixer are independent subsystems they can be multiplied to form the complete representation.

The complete representation will take the standard state space form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned}\tag{14.1}$$

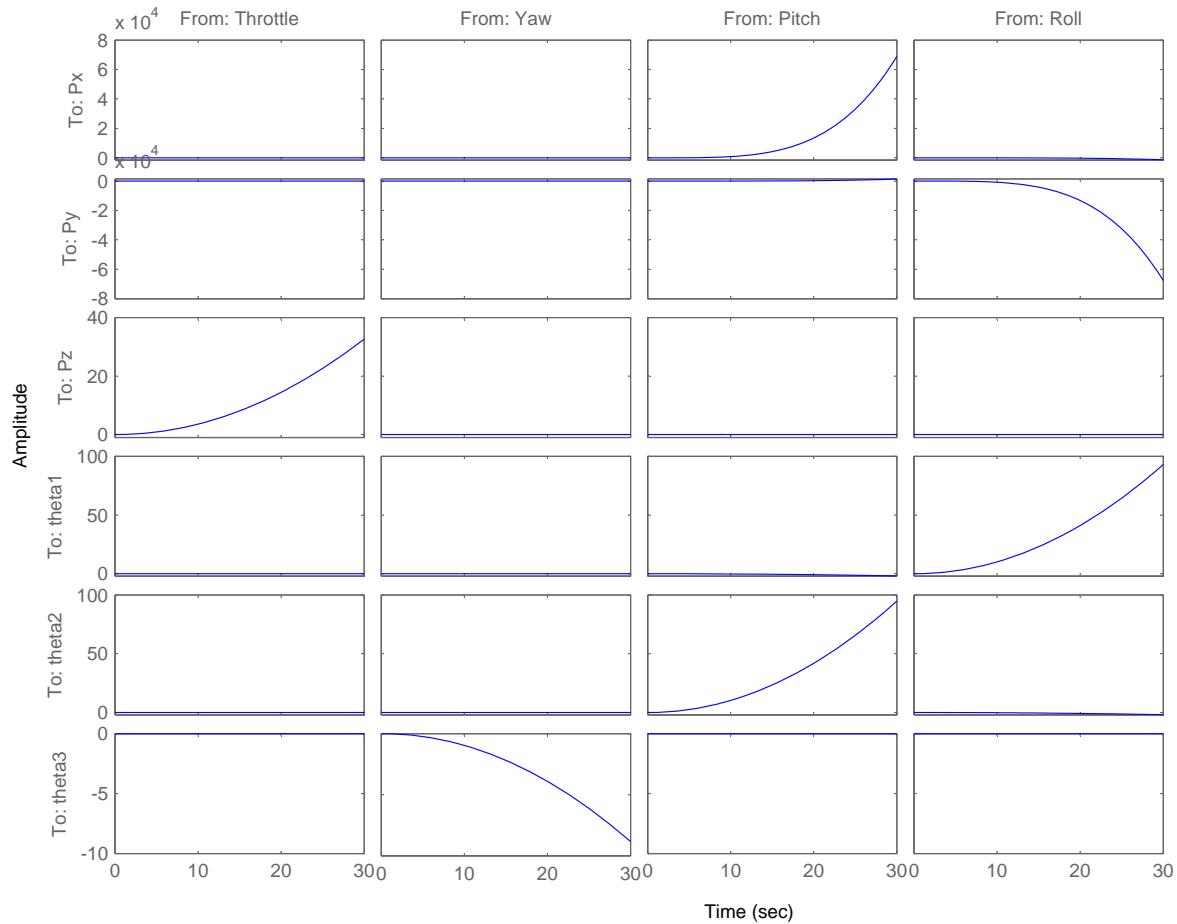
where  $x$  is the state vector,  $u$  is the input vector and  $y$  is the output vector.

A number of states in the system does not affect the input output relationship, hence the actual number of states is reduced. In the representation of the mixer, only the  $D$  matrix contains values and the states of the system are then reduced to 20. The reduction is done using the Matlab command `mssys = sminreal(sys)`.

The poles for the open loop system matrix are found to be

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -80.6289 & -6.1405 & -80.6289 & -6.1405 \\ -80.6289 & -6.1405 & -80.6289 & -6.1405 \end{array}\tag{14.2}$$

It can be seen that 12 of the poles are zero and thereby the system is marginal stable and needs to be controlled. The 12 poles also results in large values in positions and Euler angles in short time when affected by a step on throttle, pitch, yaw and roll for 30 seconds. This is shown in figure 14.1.



**Figure 14.1:** Step response on the open loop system.

The design of the controller will be done in the following 3 steps.

1. Set up the control law for the state space controller
2. Design of observer if required
3. Combining the control law and a possible observer

The controller will be based on the requirements listed in section 3.

## 14.1 Control Law for the State Space Controller

Before designing the controller for the state space model, it must be checked for controllability. This is done using the controllability matrix, which yields [GFFEN06, p. 457]

$$C = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix} \quad (14.3)$$

where  $n$  is the dimension of the state description, in this case 20. Then the inverse of a nonsingular matrix  $T$  is computed with rows  $t_n$ , where

$$t_n = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix} C^{-1} \quad (14.4)$$

Based on [GFFEN06, p. 458] this can only be done theoretically and therefore, the controllability of a real system will be calculated using a numerically stable form that converts the system matrix to a staircase form. Further calculations are done using Matlab command `[Abar,Bbar,Cbar,T,k] = ctrbf(msys.A,msys.B,msys.C);` that iteratively converts the general state description to a staircase form. The sum of the variable  $k$  is equal to the amount of controllable states and is found to be 20.

The second step in designing the controller for the system is finding the Control Law [GFFEN06, p. 471]. It is decided to construct the linear control system as an optimal Linear Quadratic Regulator (LQR). To do this, a cost function is made that minimises the performance index,  $J$ , for

$$\dot{x} = Ax + Bu \quad (14.5)$$

$$z = C_1x + Du \quad (14.6)$$

where  $z$  is the outputs that are to be controlled and  $C_1$  is the output matrix that includes the elements for  $z$ . The performance index can be written as, [GFFEN06, p. 487]

$$J = \int_0^\infty [\rho z^2(t) + u^2(t)] dt \quad (14.7)$$

where  $\rho$  is a gain factor weighting the control.

The performance index can be rewritten to a form that makes it possible to calculate a control gain that minimises the performance index in Matlab. This is shown in the following equation.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (14.8)$$

Matlab calculates the control gain  $K$  based on the  $A$  and  $B$  matrices and the 2 matrices,  $Q$  and  $R$ , diagonal matrices that describes the relationship between control and performance. The control law is defined as

$$u = -Kx = - \begin{bmatrix} K_1 & K_2 & \dots & K_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (14.9)$$

and is only valid under the assumption that all states are available.

Equation (14.9) is then substituted into equation (14.1) which yields

$$\dot{x} = Ax + B(-Kx) = Ax - BKx = (A - BK)x \quad (14.10)$$

$$y = Cx + D(-Kx) \quad (14.11)$$

The control gain,  $K$  is then calculated in Matlab using command  $K = \text{lqr}(A, B, Q, R)$ , and the implementation of the controller gain can be seen in figure 14.2.

To design  $Q$  and  $R$ , Bryson's rule is used [GFFEN06, p. 493]. The elements in the diagonal of the  $Q$  matrix scales the contributions from the state vector and the diagonal of the  $R$  matrix scales the contribution from the input signal. Bryson's rule yields that the scalars must be

$$\text{scalar} = \frac{1}{(\text{maximum acceptable value})^2} \quad (14.12)$$

The deviations will be taken from the requirement specification. After implementation of the controller, the poles for the system was found to be

$$\begin{array}{cccc} -80.6311 & -80.6304 & -80.6315 & -80.6312 \\ -5.5322 + 3.3111i & -5.5322 - 3.3111i & -5.2600 + 2.7517i & -5.2600 - 2.7517i \\ -2.0647 + 2.3782i & -2.0647 - 2.3782i & -1.9971 + 2.4091i & -1.9971 - 2.4091i \\ -5.3421 & -6.0982 & -2.8205 & -0.9973 \\ -0.9973 & -0.6129 + 0.3342i & -0.6129 - 0.3342i & -1.0917 \end{array} \quad (14.13)$$

It can be seen that all the poles are placed in the left half plane and thereby the system is stable. To be able to control the output, a reference input is introduced.

## 14.2 Introducing Reference Input

After implementing the control law it is desired to be able to control the output with a reference input. This is done by making the output only depend on the input signal in steady state. To implement the steady state, the states  $x_{ss} = N_x r_{ss}$  and the inputs  $u_{ss} = N_u r_{ss}$ , where  $N_x$  scales the state vector and  $N_u$  scales the inputs. In steady state  $\dot{x}$  is equal to zero and from the previously information the following equation can be set up [GFFEN06, p. 481].

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (14.14)$$

As it is only possible to control the position in the x, y and z axis and yaw (heading), the reference vector has the length 4. The  $C$  matrix will then be reduced to be of size  $4 \times 20$  and the  $D$  matrix to the size  $4 \times 4$ . To keep the control law active when the reference is non-zero the  $K$  matrix will be multiplied by  $N_x$ . The complete reference input matrix will then be

$$\bar{N} = N_u + KN_x \quad (14.15)$$

With the control law and reference input settled the next section will examine if all states are available. If not, the states must be estimated.

### 14.3 Observer Design

The 20 states in the system are checked for observability. This is done in Matlab using the command `[Abar, Bbar, Cbar, T, k] = obsvf(A, B, C)`, and uses the same principle as in equation (14.4), where matrix  $C$  is substituted with  $A$ . The sum of  $k$  is equal to the number of observable states, and is found to be 20.

As seen in figure 14.2, the observer is basically a copy of the system and can be described as

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L(\hat{y} - y) \\ \hat{y} &= C\hat{x} + Du\end{aligned}\tag{14.16}$$

$y$ ,  $x$  and  $\dot{x}$  from equation (14.1), are all substituted with their estimated variables  $\hat{y}$ ,  $\hat{x}$  and  $\dot{\hat{x}}$  respectively. The estimated output from the observer output is subtracted from the system output, and multiplied with the observer gain matrix  $L$ . The observer gain matrix continuously tracks the difference between  $\hat{y}$  and  $y$ , the output error, which are fed back to the observer. Variations in the state estimate error can be described as

$$\dot{\tilde{x}} = \dot{\hat{x}} - \dot{x}\tag{14.17}$$

Substituting the input vector  $\dot{\hat{x}}$  and  $\dot{x}$  from (14.16) and (14.1) respectively, into (14.17) yields

$$\dot{\tilde{x}} = A\hat{x} + Bu + L(\hat{y} - y) - (Ax + Bu)$$

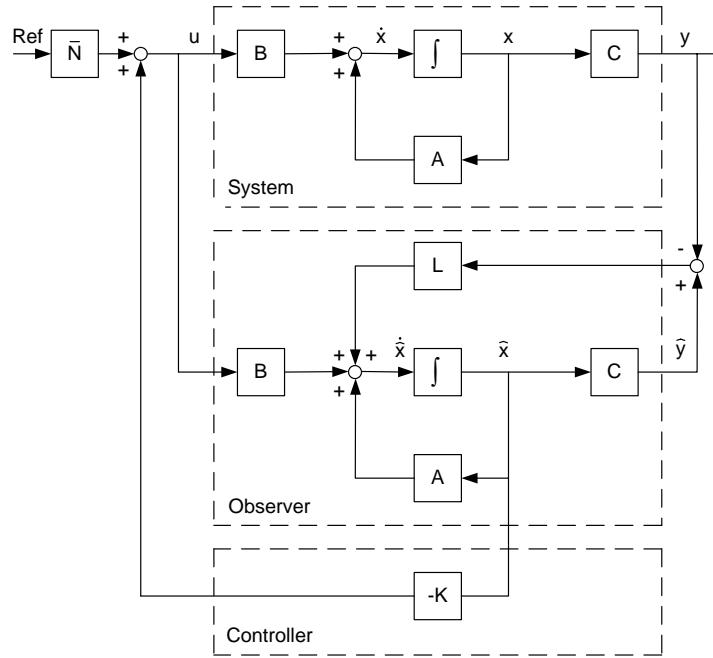
where the estimate error  $\hat{y}$  and  $y$  are substituted with the output matrix multiplied with both the state estimate and state vector.

$$\begin{aligned}\dot{\tilde{x}} &= A\hat{x} + Bu + L(C\hat{x} - Cx) - Ax - Bu \\ &= A(\hat{x} - x) + LC(\hat{x} - x) \\ &= (A + LC)(\hat{x} - x) \\ &= (A + LC)\tilde{x}\end{aligned}\tag{14.18}$$

In order to avoid conflict with the controller poles, the observer poles must be faster. If matrix  $L$  can be chosen suitable so that  $A + LC$  found in equation (14.18) has eigenvalues that are faster than the controller poles, then the small signal error  $\tilde{x}$  will converge to zero regardless of the input. A rule of thumb is to choose the observer poles to be 2-6 times faster than the controller poles, in which a factor times 4 is used. Using Matlab command `Place(A', C', KPOLES*4)` calculates the  $L$  matrix that moves the poles by a factor of 4.

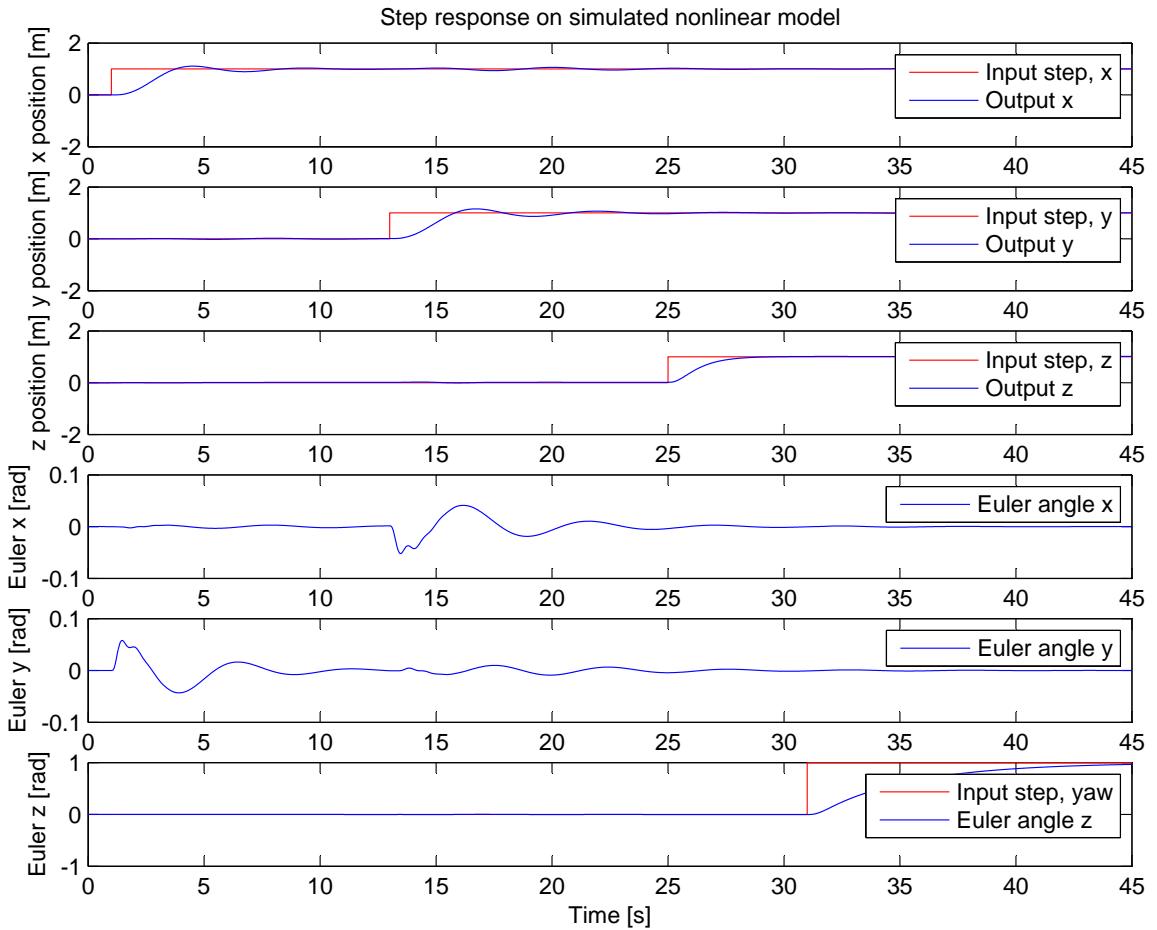
## 14.4 Combining Control Law and Observer

The next step is to combine the control law and the estimated states which can be seen in figure 14.2.



*Figure 14.2: Diagram of the system, observer and controller.*

After combining the parts, the final controller has been simulated on the non-linear model in Simulink. A step response on the 4 inputs on the model has been done and a plot can be seen in figure 14.3.



**Figure 14.3:** Step response on the simulated model. On the figure the cross coupling between the axis can be seen.

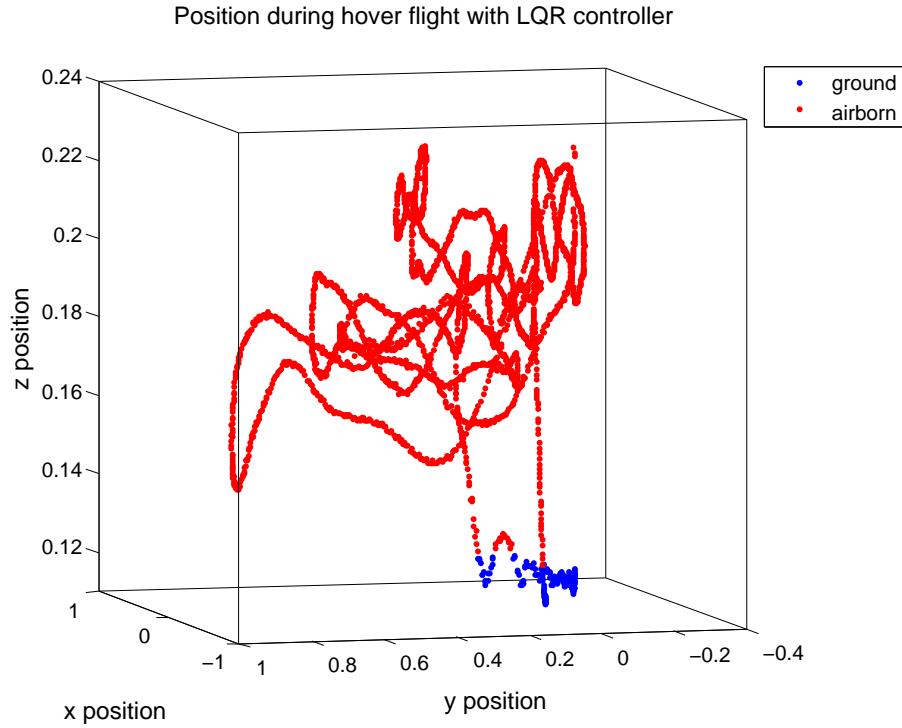
## 14.5 Testing the LQR Controller

The LQR controller will be tested in accordance with the accept test described in section 3.1 on page 14. The LQR controller is used under the same flight conditions as the PID controller. The results from the two tests are outlined in the following.

### Test 1: Keeping position and angle

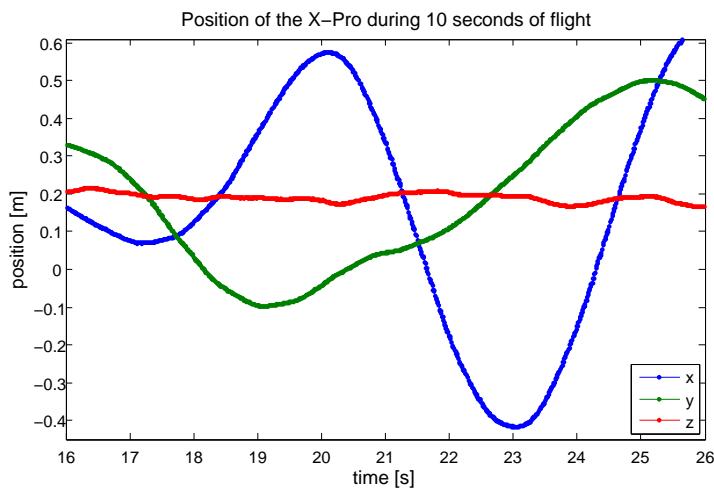
As with the PID controller the X-Pro is brought to hover and the position and angles are recorded. The complete flight data is shown in a 3D plot in figure 14.4. The LQR controller derived for this test performs worse than the PID controller as it uses more space than the PID controller. One big difference is also that the LQR controller has a steady state error on both the position and angle. The reference for the z position is set to 0.3 meters but it only reaches 0.2 meters. The x and y position is set to zero but it is a little off in both directions.

This is also shown in figure 14.5 where the position of the X-Pro during 10 seconds of flight is



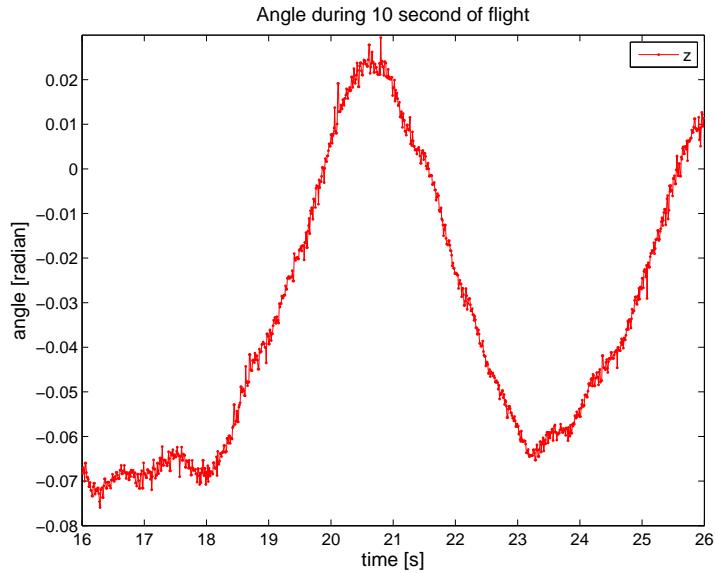
**Figure 14.4:** A 3D plot of the position of the X-Pro during the LQR hover test.

plotted. The position in the x direction varied from -0.4175 to 0.6319 meters, and the y direction varied with -0.0979 to 0.5005 meters. A video recording has been made of the flight, which can be seen on CD\video\LQR. Thus the maximum deviation from the reference which were 0 would in this case be 0.6319 meters in the x direction and 0.5005 meters in the y direction, which is well over the  $\pm 0.1$  meter defined in the requirements. The z position were also off by maximum -0.1344 meters compared to the reference of 0.3 meters. This is also over the required  $\pm 0.05$  meters.



**Figure 14.5:** The position of the X-Pro in hover with the LQR controller.

The angle around the z axis during hover is shown in figure 14.6. The performance here is better and it manages to stay within the required  $\pm 0.1$  radian. But, the angle fluctuates more than with the PID controller and there is again a clear steady state error.

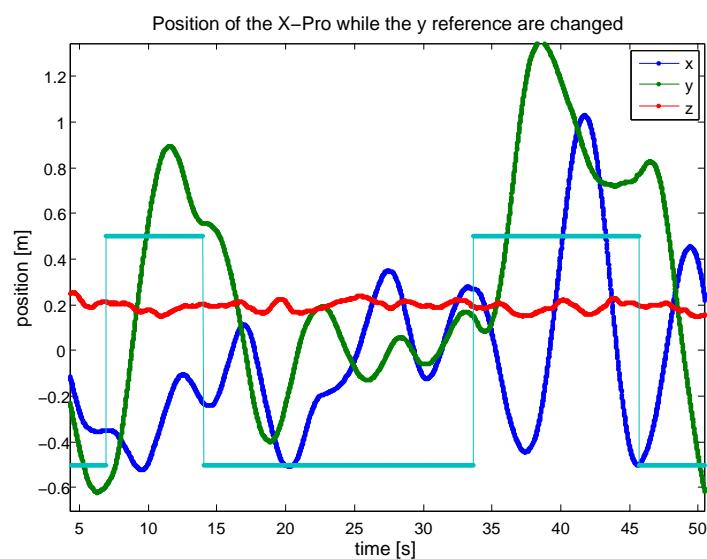


**Figure 14.6:** The angle of the X-Pro in hover with the LQR controller.

## Test 2: Changing position

In the second test the reference position was changed with 1 meter in the y direction. This has been changed from 0.5 meters in this case as the LQR controller has problems keeping the X-Pro steady. This way the changes can be seen and not just disregarded as ordinary position error. This test is performed similar to the test for the PID controller and the position during the test is shown in figure 14.7.

As it can be seen in figure 14.7 the LQR has problems staying in the correct position. The y position slowly moves away from the reference position. The LQR controller do not fulfill the requirement to be stable within 2 seconds after the reference has been changed.



*Figure 14.7: The position of the X-Pro while changing the y position with the LQR controller.*

# Chapter 15

## Controller Discussion

Two controllers for the X-Pro was designed, a PID and an LQR controller. The PID controller was designed relatively fast without use of the mathematical model. The PID controller was an easy way to get to use the Vicon system and test the signal flow in the system, as well as trying to fly with the X-Pro. The time used to fine-tune the PID controller was short, and a better tuning could have resulted in a better controller.

The LQR controller was designed to make the X-Pro hover like it did with the PID controller. Since it was a model based MIMO controller, it should be better to control the X-Pro by taking cross correlations and dynamics into account. However, the PID controller performs better than the LQR controller, which had steady state errors and large fluctuations in the position. Implementing an integral part in the LQR controller should be able to help counter the steady state error but due to lack of time this have not been done. The same goes for changing the weighting in the Q matrix, which might have a positive effect on the response.

As described in appendix D, the center of mass of the measured object is defined in the Vicon IQ software. If the center of mass defined in the software varies from the actual center of mass of the X-Pro, the position and the Euler angle measurements will deviate from the actual position and angles of the X-Pro. This adds an error to the feedback signal which the controller will try and counter, resulting in an erroneous controlsignal, thus never reaching 100% steady state. As there are insecurities about the position of the center of mass as well as the Euler angles when the X-Pro is supposed to be horizontal, this is most likely part of the control problem. It has also been observed that there is a small delay in the feedback loop, around 13 ms from controller output until the controller receives the measured change. This is inevitable considering the complexity of the measuring system, but it can have an affect on the control performance.

Another problem tuning the X-Pro is the amount of airflow that it generates with its large rotor area compared to the size of the room it is flown in. This makes for a lot of external influences on the X-Pro that have not been modelled, which makes it harder to control.

Comparing flight data from the PID and LQR controller seen in figure 13.4 and 14.4 respectively, with the manual flight data seen in figure 10.2, shows that the PID controller is the most stable. In the 3 flight tests, the aim was to maintain the X-Pro hovering. Caused by the ground effect

the manual flight data varies more than a meter in the x and y directions, when trying to keep the X-Pro hovering. The PID controller is much more stable, and only deviates with approximately 4 cm on the z-axis and approximately 20 cm on the x- and y-axis. The LQR controller deviates with approximately 8 cm on the z-axis, but is unstable in the x- and y-axis with a deviation of 1 meter.

## Part IV

### Closure



# Chapter 16

## Conclusion

The X-Pro differs from ordinary helicopters by its quadrotor design and its inability to adjust anything but the rotational velocity of these rotors. Despite of the quadrotor design, the electronics mounted on the helicopter provides ordinary helicopter control in the form of throttle, pitch, yaw and roll. To measure the position and attitude of the X-Pro, the Vicon Motion Tracking System is used providing accurate, almost real time measurements.

The goal of this project was to attain autonomous flight with the X-Pro in hover state. To achieve this, the X-Pro was modelled in order to create a modelbased controller. The model was divided into parts to reduce the complexity of the model and simplify the parameter estimation. The parts consists of the body, motor/gear and the rotors, where the first is divided into kinematics and dynamics and the two latter are part of the force generation.

The kinematics part of the modelling describes the motion of the X-Pro without considering the forces affecting it while the dynamics describes the forces requires to cause the motion. This provides a mapping of how the X-Pro responds when affected by a force in the form of a rotational matrix, describing how the coordinate system of the body turns and moves compared to the universal coordinate system.

The rotors generate the lift and drag forces affecting the X-Pro. The characteristics of the rotors were found experimentally, by measuring the forces generated. A second order polynomial describes the correlation between the rotational velocity of the rotors and the force generated, although the limited measuring options provides some insecurity about the accuracy of the values.

The motors and the attached gearing drives the rotors based on a PWM input signal from the mixer. The transfer function of the motors was found using system identification. A second order ARX model has been chosen providing an 87% fit to the measured data with the estimated parameters.

The combined model was non-linear, so in order to use a linear controller, the model was linearised. This was done using the hover position and hover attitudes as operating point. The linearised model was compared to the non-linearised model, and the linearised model provided a satisfactory response.

Two controllers were designed to control the X-Pro. First an ordinary PID controller and then a model based LQR controller. The PID controller was made using trial and error to fine tune the parameters necessary to make the X-Pro hover. The PID controller have an inner attitude control loop and an outer position control loop. The inner attitude control loop, consisting of merely a proportional gain, uses the output of the position loop as reference. The PID controller is capable of keeping the X-Pro in hover, and while the heading and the altitude are within the boundaries specified in the requirement specification, both the x and y position are not. The X-Pro floats outside of the desired area, and while the controller attempts to bring it back, it does not happen fast enough, resulting in steady movement around the reference point. Comparing the flight data obtained by manual flight, with flight data using the PID controller, both data sets representing attempts to stay in hover, the PID controller is better at keeping the X-Pro in hover.

A few key points can be improved to get a better PID controller. The most immediate improvement is a better Vicon model representation. Any misplacement of the center of mass results in the Vicon reference signal being off for the position whenever the X-Pro is not 100% in hover. And if the X-Pro representation in the Vicon Synstem in hover state does not correspond exactly to the hover position of the X-Pro, the measured Euler angles will be incorrect. More tuning of both attitude and position control is also needed. The attitude control can be changed to provide a faster response in order to make the X-Pro react faster, by changing the gain and/or changing the saturation limiting the attitudes the X-Pro can obtain. The position controller for the x and y position should also be fine tuned more to increase rise time and decrease settling time.

The modelbased LQR controller was designed based on the state space representation of the X-Pro, containing an observer and a controller. The LQR controller does not work as well as the PID controller when attempting to keep the X-Pro in hover position. The positions are fluctuating a lot around the hover point in all 3 directions and it never settles. Neither of the controllable states are within the limitations stated in the requirement specifications except for yaw, and although it is within the boundaries specified, it is still fluctuating more than the PID controller.

In order to optimise the LQR controller, an integral part should be added to counter the obvious steady state errors. In order to counter the fluctuations in position, the Q matrix needs to be weighted differently.

# Chapter 17

## Perspective

The X-Pro is capable of indoor hovering, monitored by the Vicon Motion Tracking System. This allows precise measurements of the position and Euler angles, used in both the PID- and LQR controller. However, it prevents the X-Pro from flying outdoor. Besides mounting a battery on the X-Pro, flying outdoor requires extra sensors added, such as GPS and IMU, in order to obtain position and angles.

The linear model that was developed, is based on the assumption that the X-Pro is only hovering. External influences such as wind gust affecting the X-Pro during flight, has not been taken into consideration. Flying outdoor extends the requirements of the controller as the X-Pro will often deviate from the operating point, thus resulting in the need for e.g. a non-linear controller. It is also preferred to implement the controller onboard the X-Pro, to eliminate the control data transmission between a ground station and the X-Pro, thus eliminating the risk of losing the control signals. The only thing sent from the ground could be new position references.

Unmanned aerial vehicles can be small and agile, providing huge mobility in cities where ordinary scale aircrafts can't manoeuvre. Autonomous flight gives the ability to reach destinations humans can not see or reach as well as reducing the need for human interaction. In combination, it gives the possibility of large area or crowd surveillance with very little need for human interaction. As such, a model similar to the X-Pro has been used for spectator surveillance at German football stadiums, and while this was done with human control, autonomous control can provide control of several UAV's without an increase in the needed manpower.



# Bibliography

- [06g06] Group 06gr831. Draganflyer x-pro modelling and control. Department of Control Engineering, 2006.
- [Bak02] Thomas Bak. Lecture notes - modeling of mechanical systems. Department of Control Engineering, 2002.
- [BlCH08a] Morten Bisgaard and Anders la Cour Harbo. Motion tracking laboratory. Department of Electronic Systems Section for Automation and Control Aalborg University Fredrik Bajers Vej 7, 9220 Aalborg East, Denmark, 2008.
- [BlCH08b] Morten Bisgaard and Anders la Cour Harbo. Simulink block for real time streaming. [http://es.aau.dk/projects/robotics/labs/motion\\_tracking\\_lab/simulink\\_block/](http://es.aau.dk/projects/robotics/labs/motion_tracking_lab/simulink_block/), 2008.
- [CHE02] David E. Penney C. Henry Edwards. Calculus, 6'th edition. Prentice Hall, 2002.
- [Cra89] John J. Craig. *Introduction to Robotics Mechanics and Control*. Addison-Wesley Publishing Company, Inc, 2nd edition, 1989.
- [GFFEN06] J. David Powell Gene F. Franklin and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson Prentice Hall, Upper Saddle River, New Jersey 07458, 5th edition, 2006.
- [Gui07] Study Guidance. [http://esn.aau.dk/fileadmin/esn/Studieordning/Cand\\_SO\\_ed\\_aalborg\\_maj08.pdf](http://esn.aau.dk/fileadmin/esn/Studieordning/Cand_SO_ed_aalborg_maj08.pdf), 2007. Master 7.-10. semester, Electronic and Electro technique along with Data technique.
- [Ins08] National Instruments. Ni pci-6071e. [http://www.ni.com/pdf/products/us/4daqsc199-201\\_ETC\\_212-213.pdf](http://www.ni.com/pdf/products/us/4daqsc199-201_ETC_212-213.pdf), 2008.
- [Vic08] Vicon. A complete 8-camera vicon system at a great price. <http://www.vicon.com/spots/MX3systematagreatprice.html>, 2008.

## Bibliography

---

# Part V

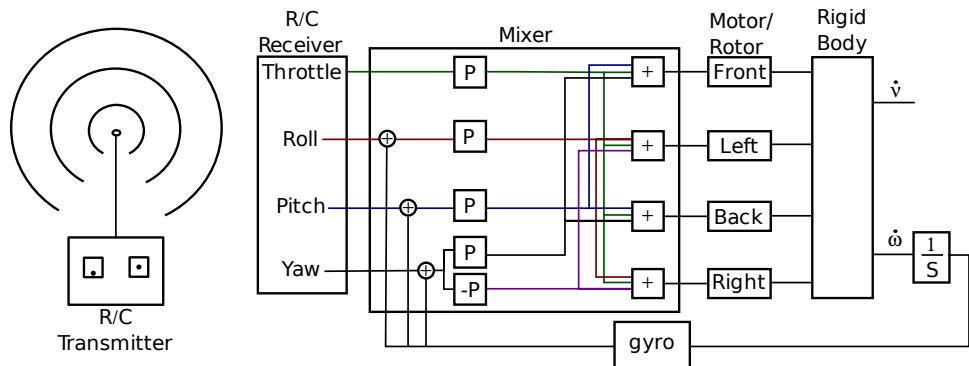
# Appendix



## Appendix A

Analysis of the X-Pro hardware

The X-Pro includes electronics that enables control of the X-Pro like an ordinary helicopter<sup>1</sup> and to stabilize it while flying. This appendix will describe the electronics divided in, R/C transmitter, mixer and gyroscope, which is shown in figure A.1.



**Figure A.1:** Illustration of the signal flow from the R/C transmitter to the rigid body.

The R/C transmitter is the device used for manually control of the X-Pro, the gyroscope measures the angular velocities, which are scaled and looped back to the mixer and the mixer translates the received R/C signal to PMW signals for the motors.

### A.1 R/C Transmitter

This section describes 2 parts involving the R/C. The first part describes the signal sent from the R/C transmitter to the mixer and the next part describes the R/C settings that are used in this project.

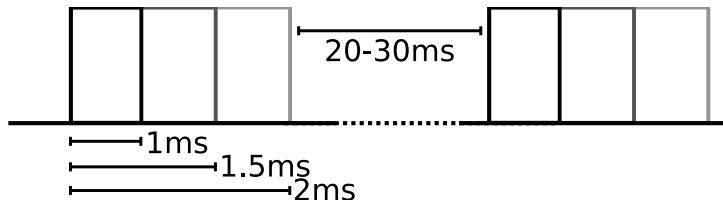
---

<sup>1</sup> Helicopter with one horizontally mounted main rotor and a vertically mounted tail rotor.

### A.1.1 R/C Signals

The signal transmitted from the radio controller are formed as a pulse width modulated signal. The pulse time of the PWM signal indicates the amount of control. The pulse time can vary from 1 ms to 2 ms and represents minimum and maximum respectively.

An example of a signal from an R/C transmitter is shown in figure A.2.



**Figure A.2:** Illustration of a pulse width modulated signal send by an R/C transmitter. The Illustration shows 3 pulses representing 0%, 50% and 100 % in different graytones.

The R/C transmitter sends 4 different signals as described in the analysis. These signals are throttle, yaw, pitch and roll. To make the X-Pro act like a normal helicopter using these 4 signals, they have to be mixed according to the rotational velocity described in the analysis. The outputs from the mixer are 4 PWM signal that controls the speed of the motors only by varying the duty cycle. The signal flow from the R/C receiver on the X-Pro to the PWM outputs are shown in figure A.1.

After the analysis of the R/C signals some settings of the R/C transmitter will be examined.

### A.1.2 R/C Settings

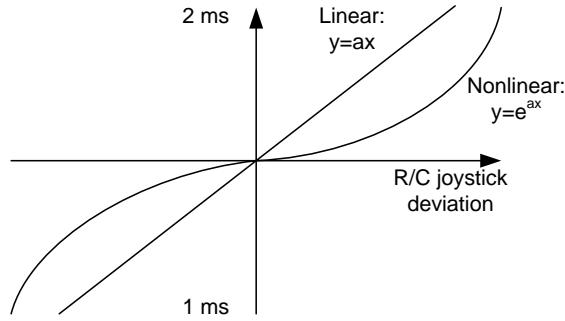
Before identifying the relationship between the R/C transmitter and the PWM output it is important to understand in which way the R/C transmitter behaves as it will be used to send out the signals for autonomous flight.

The signals from the R/C transmitter can be send linear and non-linear. The non-linear way makes the joystick less sensitive around the center and more sensitive away from the center. This option is used to make it easier for manually control of the X-Pro. An illustration of a non-linear signal is sketched in figure A.3.

The nonlinear function is expressed as  $y = e^{ax}$ , which is a normal exponential function. This non-linearity is unwanted when controlling it from the interface PC to make mapping easier to implement. By setting the R/C transmitter to send signals linearly, manually flight will become more difficult and autonomous flight of the X-Pro does not require high resolution near the joystick centers. An illustration of a linear signal is also sketched in figure A.3.

Another often used setting is the trimming of the signals, which will result in a offset of the signal that are sent, The offset will also be included in the model.

It is observed that some of the signals send are inverted by the R/C transmitter. To understand the mapping of the signals, they are all measured and written in table A.1. The ConLink block mentioned in the table is further clarified in chapter E.2.



**Figure A.3:** linear and non-linear R/C transmitter settings.

Signal	ConLink Send	Received	PWM front	PWM back	PWM left	PWM right
Throttle	↑	↓	↓	↓	↓	↓
Yaw	↑	↑	↑	↑	↓	↓
Pitch	↑	↓	↑	↓	→	→
Roll	↑	↓	→	→	↑	↓

**Table A.1:** This table shows the relationship between what is send through to the ConLink Simulink module and what is received by the R/C receiver connected to the interface PC, simultaneously. For instance, if the throttle signal send to the ConLink module is increased the received throttle signal decreases, and so does the PWM signal on all the motors.

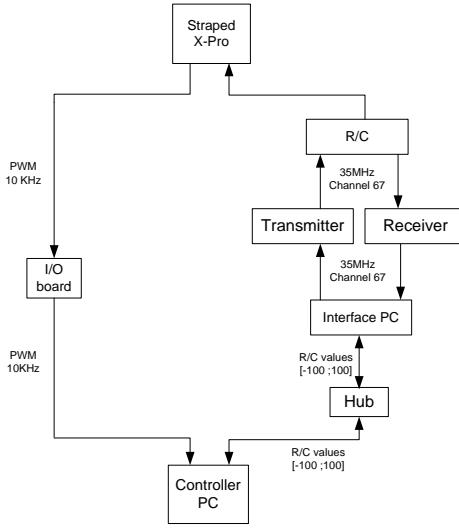
The R/C transmitter also has a trainer switch which is used to switch between manually flight and autonomous flight.

## A.2 Mixer

The mixer is an electronic circuit that splits the received R/C signals to make 4 control signals for the motors. It is experienced by tests that the 3 signals, yaw, pitch and roll only have influence if the throttle is above a certain threshold. Furthermore values of manipulating yaw, pitch and roll are added to the value from of the throttle. The signal flow can also be seen in figure A.1.

To identify the scaling between the input and the output of the mixer, the signal from the R/C transmitter and the PWM signal from the mixer must be obtained. As seen in figure A.4 the R/C transmitter signals are logged in the Interface PC through a receiver which is connected through USB to the Interface PC.

This signal is then sent to the Controller PC through an Ethernet connection. Furthermore, 4 probes are connected to measure each of the PWM signals on the motors and these measurements are also sent to the Controller PC. The measuring of the PWM signals on the motors are done using a National Instruments 6071 I/O board. A short description of the I/O board can be seen in section B.1.



**Figure A.4:** Signal flow for PWM and R/C transmitter measurements.

### A.2.1 Test description

The first test is based on changing the the R/C signal for throttle. The maximum range of the signals are [-100;100] and in the test it is changed from -90 to 90. After this the throttle is set back to a level to get a PWM period on the motors of approximately 50% and then change yaw, pitch and roll from 0 to -90 and then to 90 and back to 0.

To support this a Simulink model is made were a sinusoidal signal is generated on 1 channel while keeping the other channels steady. The corresponding PWM signal from 1 motor, and the send and -received R/C signal are then logged. The Simulink model can be found on the cd, CD\models\mixer\mixer\_yaw\_test.mdl.

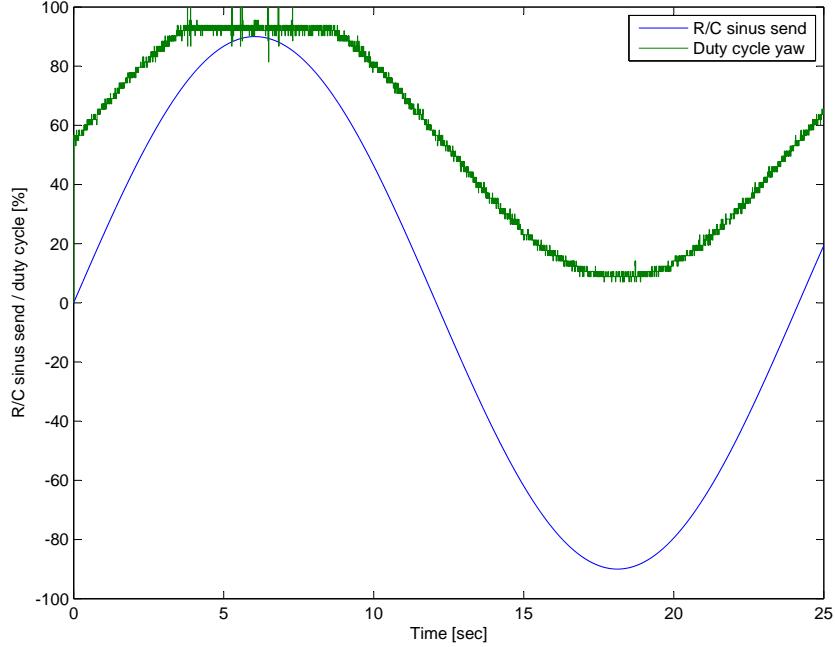
### A.2.2 Test Result

The result from the test were a set of signals that shows how throttle, yaw, pitch and roll affects the PWM signal. In figure A.5 the result from the yaw test is shown. The other figures for throttle, pitch and roll can be found using an m-file on the cd , CD\m-files\mixer\split\_pwm\_mess\_signal.m.

It is assumed that the gain for throttle, yaw, roll and pitch is the same for all four motors, and therefore only the front motor is examined. It is experienced that the gains can be fitted by a first order polynomial.

The first order polynomial fit is found by the Matlab command `polyfit(x,y,1)`, and the result from this can be seen in table A.2.

The identified gains are added to the model. The next thing is to find the feedback from the gyroscope.



**Figure A.5:** Plot of PWM on a motor and R/C transmitter measurements.

Signal	Throttle	Yaw	Pitch	Roll
Gain	-0.54	0.53	0.52	-0.51

**Table A.2:** Identified gains for throttle, yaw, pitch and roll.

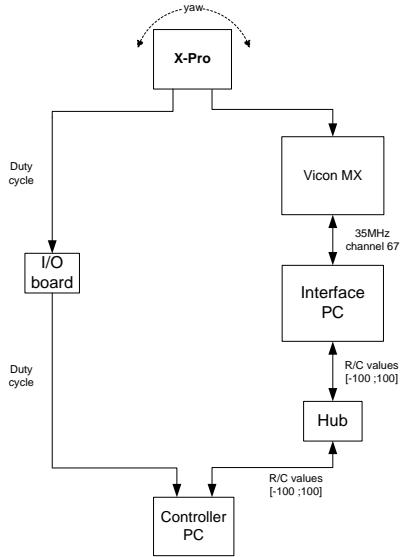
## A.3 Gyroscope

To stabilize the X-Pro a gyroscope on the X-Pro measures the angular velocity and makes a feedback signal to the mixer in order to compensate for unwanted maneuvers and to aid in the control of the X-Pro. To include the gyroscope effect in the model of the X-Pro, the amount of feedback gain must be identified. The feedback gains are illustrated in figure A.1.

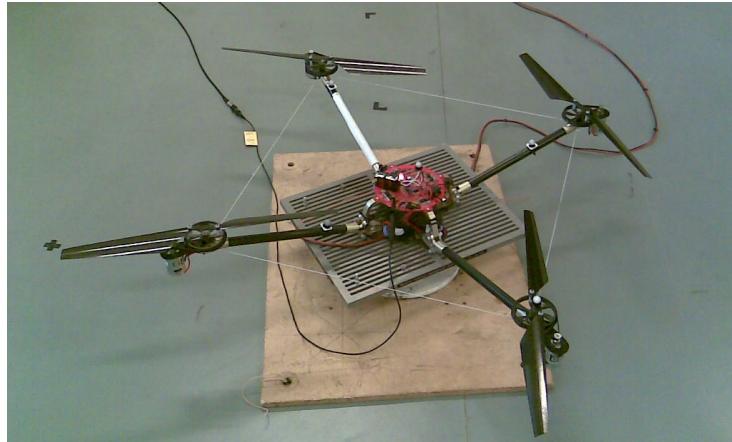
### A.3.1 Test description

To identify the gains caused by the gyroscopes an experiment has to be done. The idea is to use the Motion Tracking System to track the x,y and z positions and angles of the X-Pro, while manually making the X-Pro yaw, pitch and roll. Differentiating the angles derives the angular velocities of the X-Pro in the 3 directions.

By mapping the euler angles from the Motion Tracking System to the duty cycle for the motors it is possible to identify the feedback from the gyroscope. The test setup can be seen in figure A.6, where a probe is connected to one of the motors which is connected to the I/O board in order to measure the duty cycle that controls the motor.



**Figure A.6:** Signal flow for duty cycle and Vicon measurements.



**Figure A.7:** image of the gyro testsetup for the yaw angle.

The feedback are measured for pitch, roll and yaw. Yaw is the easiest to measure, as it can be placed on a turning lathe as seen in figure A.7. It is assumed that the yaw gain is equal for all the motors just with a sign change on front and back compared to the left and right motors as the rotates in different directions.

An amount of throttle is needed before the X-Pro reacts on pitch, roll and yaw -input and therefore, the joystick position for throttle was set to approximately 30 % while doing the tests.

### A.3.2 Test results

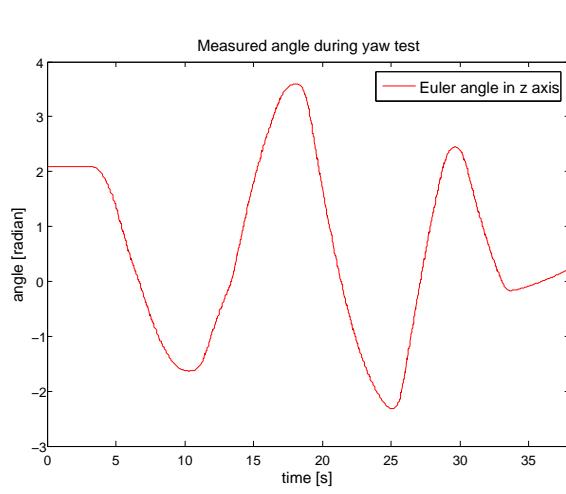
The sampling rate on the I/O board was set to 10 kHz and the Vicon Motion Tracking System operates at 100 Hz. In order to compare the measured signals, the PWM signal from the motor has to be converted to a duty cycle value. The angle measurement from the Vicon system also have to be cleaned up, as it is sampled at 10 kHz, but only have a new sample with 100 Hz. Therefor, all successive samples with the same value are removed and the signal are then differentiated to get the angular velocity. The differentiated signal contain some unwanted noise and therefore the signal are smoothed by running a 3 sample mean window over the signal.

From the test it was experienced that yaw, pitch and roll have almost the same gyroscope gain. Only the results from the yaw and roll test are included here.

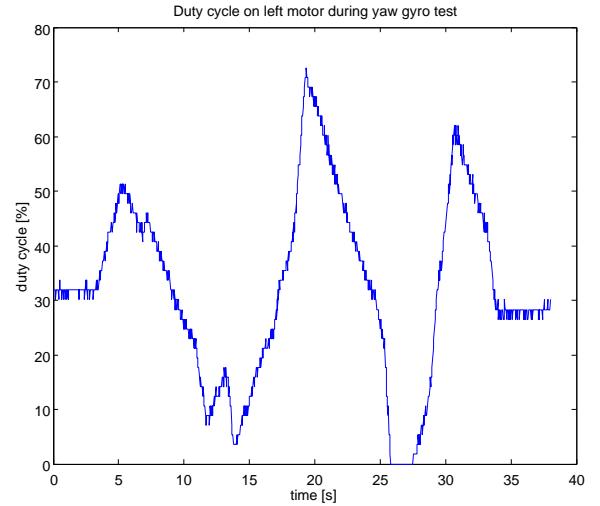
The Yaw test were performed on the turning lathe which made it possible to make long movements with constant speed. The measured euler angle is shown in figure A.8 and the corresponding PWM duty cycle is shown in figure A.9.

The differentiated angle from the yaw test is shown in figure A.10 together with the mean of the signal.

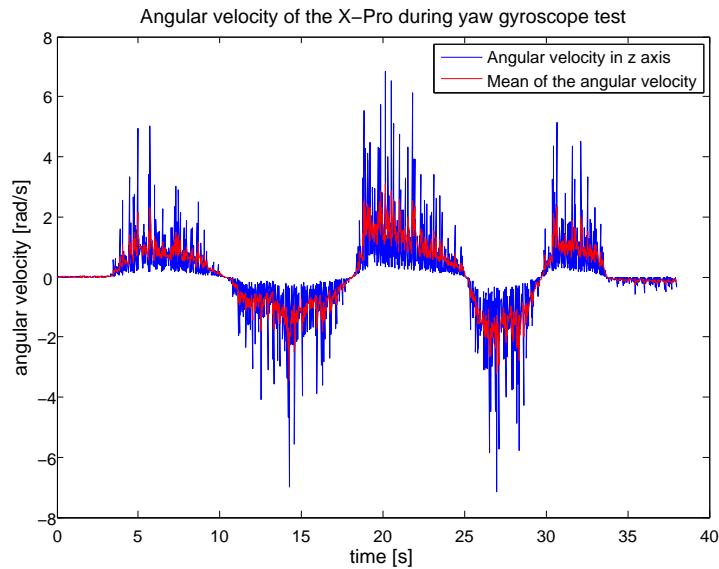
The angular velocity for yaw where compared to the change in PWM duty cycle by use of the



**Figure A.8:** Measured euler angle in the z axis during manually yaw.



**Figure A.9:** Corresponding PWM signal during yaw.

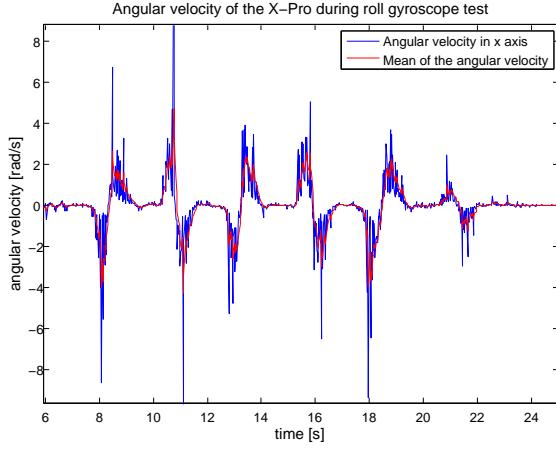


**Figure A.10:** The calculated derivative of the angular velocity and the mean of the signal.

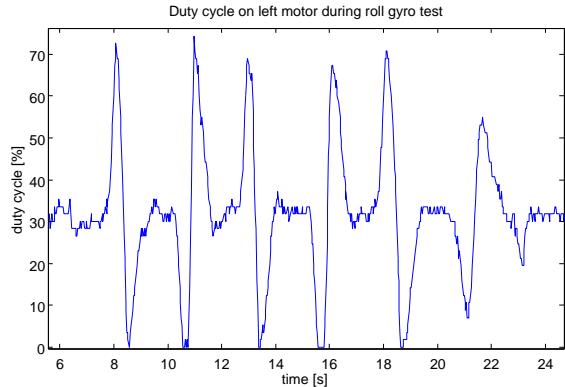
`fit(x,y,'POLY1')` in Matlab. This command fits a first order polynomial  $y = p_1 \cdot x + p_2$  to the input and returns  $p_1$  with a 95% confidence bounds. The result for yaw were  $p_1 = -16.03$  with a  $\pm 0.35$  confidence interval.  $p_2$  is of no interest in this case.

The roll test were performed by placing the X-Pro on a round cylinder, which had a 15 cm radius, and use it as a platform to easier do the roll movement with the X-Pro. This did not give as good results as the yaw test, and the movements are not as long. It was also harder to keep the other angles steady while doing the roll movement, so the result will not be as precise as with the yaw test.

In figure A.11 the angular velocity in the x axis and the mean are plotted and in figure A.12 the corresponding duty cycle are shown.



**Figure A.11:** Measured angular in the x axis during manually roll.



**Figure A.12:** Corresponding PWM signal during roll.

Again the relationship are found by using the Matlab function `fit`. The result were  $p_1 = -13.99$  with a  $\pm 0.36$  confidence interval.

The m-file used can be found on the cd, CD\m-files\gyroscope\split\_signal.m. The gain for pitch were roughly the same as roll and yaw. It is decided to use the same gyro gain for yaw, pitch and roll as the measurement of the roll and pitch gains where less precise, as not only one angle were affect while doing the test. The identified feedback gain is chosen to be -15 and added to the model.

# Appendix B

## Data Logging for Motor Estimation

This appendix will describe how the data used for the system identification on the motor including the gear, will be logged. It is assumed that the four motors as well as the four rotors are alike and under that assumption, data logging will only take place on the front motor.

### B.1 Logging the Data

To measured the velocity from the X-Pro to Matlab a computer with an installed I/O board are used. The I/O board is a PCI-6071E from National Instruments, [Ins08] and the data are logged as analog voltage.

The input signal, which is a PWM signal is a result of manually manipulation of the throttle on the R/C transmitter.

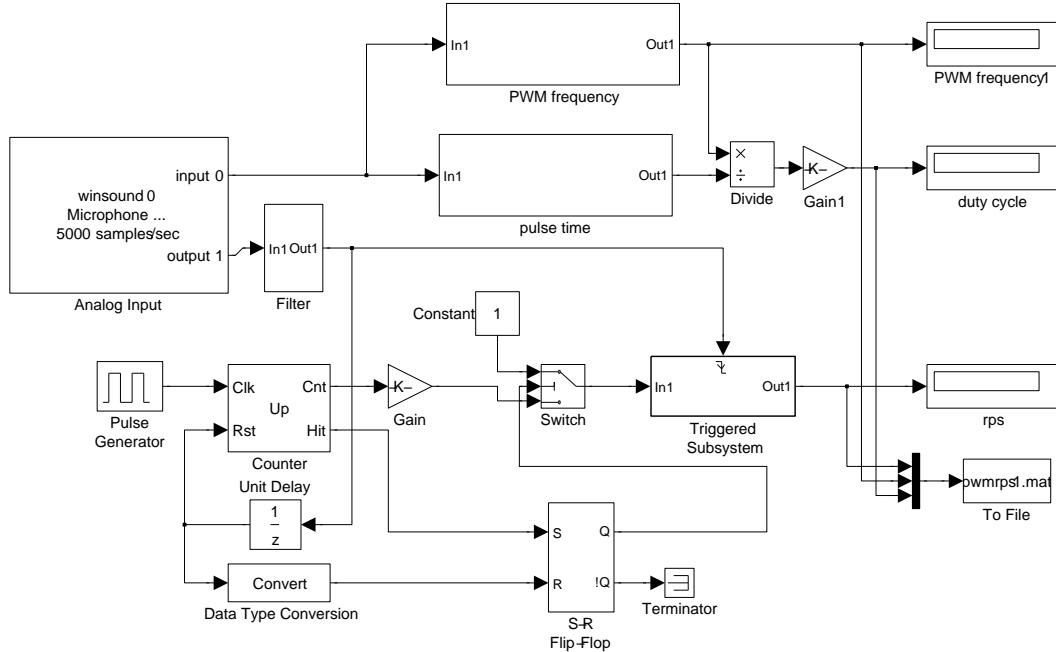
To set the minimum sampling rate on the I/O board the pulse time of the input and output signals are calculated. It is assumed that the duty cycle of the input signal for the motor will be between 50 % and 100 %. The frequency of the input signal has been measured to be around 200 Hz. In that case a pulse will be present between 2,5 ms and 5 ms.

For the sensor, measurements has proofed that the pulse is high at around 91,7 % and preliminary tests has shown that the rotational velocity will be around 25 revolutions per second. At this velocity  $\pm 5$ , the pulse from the hall sensor will be present between 36,7 ms and 45,9 ms.

From these calculations it is obviously that the sampling rate will depend on the PWM signal. To be able to follow the measured data on the pc the sampling rate of the I/O board are set to 5000 Hz. At this frequency a sampling will happen every 0,2 ms. At 50 % duty cycle it would theoretically possible to sample 12,5 times. Practically it will be 12 times and in worst case 11 times. This leads to a possible fault of 8,33 %. At 100 % the worst case will be 24 out of 25 samples, which is 4 %. Test flights has shown that the throttle will be at the upper level and the fault is tolerated and so are the sampling rate of the I/O board.

## B.2 Logging the PWM Signal

To collect the PWM signal, which is the input for the motor, a probe is connected to the input of the output stage for the motor. In Simulink a model are used to translate the logged data into useful data for SID. The model can be seen in figure B.1 and can also be found on the cd, CD\models\motor\hallsensor.mdl.



**Figure B.1:** Simulink model used for logging revolutions per seconds as a function of PWM. The model is inspirerered by previously work [06g06].

The model is made with inspiration from previously work [06g06] and additions are made to make it useful for this project.

Briefly the procedure is that a clock counts as long as a pulse is present. At the falling edge of e pulse, the amount of clock pulses are stored. In the same way, the period time are counted in clock pulses from one rising edge to the next. The pulse time divided with the period time represents the PWM.

## B.3 Logging the Revolutions

To collect the output data which is the revolutions of the rotor, a pre-installed hall sensor is used. The hall sensor is mounted under the gear wheel and a magnet is mounted on the gear wheel as in figure B.2. The output of the hall sensor is 5 V and when the magnet passes through the hall sensor, an 0 V pulse is present at a wire.

The pulses from the hall sensor are collected via the PCI-6071E I/O board from National Instruments [Ins08], as analog voltage.



**Figure B.2:** In the middle of the picture the hall sensor can be seen mounted under the gearwheel. The magnet is the white dot on the gear wheel.

To calculate the revolutions per second, the same procedure are used as for the PWM signal. The amount of clock pulses from one rising edge to the next, represents the rps.



# Appendix C

## Model Linearisation

The equations detailing the dynamics and the kinematics have been found in chapter 7 and 8 and the forces and torques are derived in chapter 6. To be able to use the equations for a linear controller the equations must be linearised.

### C.1 Method

In order to linearize the equations, first order Taylor approximation are used. The variables will be substituted with a steady state point also known as an operating point and a small-signal gain. The operating point will be described in the next section.

In the cases a first order Taylor approximation with 1 variable is used, the approximation is given as [CHE02, p. 707]

$$f(x) \approx f(\bar{x}) + \frac{\partial f(x)}{\partial x} \Big|_{x=\bar{x}} \tilde{x} \quad (\text{C.1})$$

In the cases an equation has 2 variables, the approximation is given as

$$f(x, y) \approx f(\bar{x}, \bar{y}) + \frac{\partial f(x, y)}{\partial x} \Big|_{x=\bar{x}} \tilde{x} + \frac{\partial f(x, y)}{\partial y} \Big|_{y=\bar{y}} \tilde{y} \quad (\text{C.2})$$

In order to do the linerisation some mathematical rules are needed. The first rule is concerning a cross product. The cross product can be rewritten to the following :

$${}^B\vec{\omega} \times {}^B\vec{v} = S({}^B\vec{\omega}) {}^B\vec{v} \quad (\text{C.3})$$

where  $S$  is a skew symmetric matrix and can be written as

$$S = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (\text{C.4})$$

Concerning a product of a skew symmetric matrix and a vector the following rule apply :

$$S(\vec{a})\vec{b} = -S(\vec{b})\vec{a} \quad (\text{C.5})$$

Before linearising the equations, the operating points must be found. This will be done in the following section.

## C.2 Operating Point

The operating point for the X-Pro is hover, and the controller should keep the X-Pro hovering. For the operating point a set of positions are needed for the velocity, body acceleration, angular velocity and position.

When the X-Pro is hovering the linear velocity, angular velocity and Euler angles should all be zero, which lead to the following

$$\begin{aligned} {}^I\vec{v}_B &= \vec{0} \\ {}^B\vec{\omega} &= \vec{0} \\ \vec{\theta} &= \vec{0} \end{aligned}$$

Regarding the position of the X-Pro, the objective is hover. A position just needs to be picked in order to give the X-Pro enough altitude to provide a good margin for error. As for the x- and y-axis, the center of the room would be preferable, and for the z-axis a height of 0.2 meters is chosen.

$${}^I_B\vec{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.2 \end{bmatrix} \quad (\text{C.6})$$

## C.3 Linearisation of the Body

The following 4 equations needs to be linearised.

The first equation is the derivative of the velocity of the body.

$${}^B\dot{\vec{v}} = {}^B\vec{F}\frac{1}{m} - {}^B\vec{\omega} \times {}^B\vec{v} \quad (\text{C.7})$$

The second equation is the derivative of the angular velocity of the body.

$${}^B\dot{\vec{\omega}} = {}^I{}^{-1} {}^B\vec{\tau} - {}^I{}^{-1} ({}^B\vec{\omega} \times {}^I{}^B\vec{\omega}) \quad (\text{C.8})$$

The third equation is the derivative of the Euler angles, the Euler rates.

$$\dot{\vec{\theta}} = {}^I{}^B R \vec{\omega} \quad (\text{C.9})$$

The fourth equation is the derivative of the position of the body in space.

$${}^I_B \dot{P} = {}^I_B \vec{v}_B = {}^I_B R^B \vec{v} \quad (\text{C.10})$$

Now the equations will be linearised according to the methods described in section C.1. Under the linearisation the vector and frame indexes in the equations will be neglected, to make the equations easier to read.

### Linear Acceleration

First the cross product is rewritten

$${}^B \dot{\vec{v}} = {}^B \vec{F} \frac{1}{m} - {}^B \vec{\omega} \times {}^B \vec{v} \quad (\text{C.11})$$

$${}^B \dot{\vec{v}} = {}^B \vec{F} \frac{1}{m} - S({}^B \vec{\omega}) {}^B \vec{v} \quad (\text{C.12})$$

The 1. order Taylor approximation is

$$\begin{aligned} \dot{v}(F, \omega, v) &\approx \dot{v}(\bar{F}, \bar{\omega}, \bar{v}) + \left. \frac{\partial \dot{v}(F, \omega, v)}{\partial F} \right|_{F=\bar{F}} \tilde{F} + \left. \frac{\partial \dot{v}(F, \omega, v)}{\partial \omega} \right|_{\omega=\bar{\omega}} \tilde{\omega} + \left. \frac{\partial \dot{v}(F, \omega, v)}{\partial v} \right|_{v=\bar{v}} \tilde{v} \\ &= \frac{\bar{F}}{m} - S(\bar{\omega}) \bar{v} + \left. \frac{\partial \left( \frac{F}{m} - S(\omega)v \right)}{\partial F} \right|_{F=\bar{F}} \tilde{F} + \left. \frac{\partial \left( \frac{F}{m} - S(\omega)v \right)}{\partial \omega} \right|_{\omega=\bar{\omega}} \tilde{\omega} \\ &\quad + \left. \frac{\partial \left( \frac{F}{m} - S(\omega)v \right)}{\partial v} \right|_{v=\bar{v}} \tilde{v} \\ &= \frac{\bar{F}}{m} - S(\bar{\omega}) \bar{v} + \frac{1}{m} \tilde{F} + S(\bar{v}) \tilde{\omega} - S(\bar{\omega}) \tilde{v} \end{aligned} \quad (\text{C.13})$$

### Angular Acceleration

The equation for the angular acceleration is

$${}^B \dot{\vec{\omega}} = I^{-1} {}^B \vec{\tau} - I^{-1} ({}^B \vec{\omega} \times I {}^B \vec{\omega}) \quad (\text{C.14})$$

The equation has 2 input arguments,  $\vec{\tau}$  and  $\vec{\omega}$ .

The linearised equation will be

$$\begin{aligned}
\dot{\omega}(\tau, \omega) &= \dot{\omega}(\bar{\tau}, \bar{\omega}) + \frac{\partial \dot{\omega}(\tau, \omega)}{\partial \tau} \Big|_{\tau=\bar{\tau}} \tilde{\tau} + \frac{\partial \dot{\omega}(\tau, \omega)}{\partial \omega} \Big|_{\omega=\bar{\omega}} \tilde{\omega} \\
&= I^{-1} (\bar{\tau} - S(\bar{\omega})I\bar{\omega}) + \frac{\partial (I^{-1}\tau - I^{-1}S(\omega)I\omega)}{\partial \tau} \Big|_{\tau=\bar{\tau}} \tilde{\tau} + \frac{\partial (I^{-1}\tau - I^{-1}S(\omega)I\omega)}{\partial \omega} \Big|_{\omega=\bar{\omega}} \tilde{\omega} \\
&= I^{-1} (\bar{\tau} - S(\bar{\omega})I\bar{\omega}) + I^{-1}\tilde{\tau} - I^{-1} \left( \underbrace{\frac{\partial(S(\omega))}{\partial \omega} I\omega \Big|_{\omega=\bar{\omega}}}_{\Delta} + S(\omega) \frac{\partial(I\omega)}{\partial \omega} \Big|_{\omega=\bar{\omega}} \right) \tilde{\omega} \quad (\text{C.15})
\end{aligned}$$

The part marked with a  $\Delta$  can be rewritten to

$$\frac{\partial(S(\omega))}{\partial \omega} I\omega \Big|_{\omega=\bar{\omega}} \Rightarrow \frac{\partial(S(\omega))}{\partial \omega} I\bar{\omega} \Big|_{\omega=\bar{\omega}} \Rightarrow \frac{\partial(-S(I\bar{\omega}))}{\partial \omega} \omega \Big|_{\omega=\bar{\omega}} \Rightarrow -S(I\bar{\omega}) \quad (\text{C.16})$$

The final equation for the angular acceleration ends up with

$$\dot{\omega}(\tau, \omega) = I^{-1}(\bar{\tau} - S(\bar{\omega})I\bar{\omega}) + I^{-1}\tilde{\tau} - I^{-1}(-S(I\bar{\omega}) + S(\bar{\omega})I)\tilde{\omega} \quad (\text{C.17})$$

## Euler Rates

The derivative of the Euler angles are

$$\dot{\vec{\theta}} = {}^I_B R^B \vec{\omega} \quad (\text{C.18})$$

The equation takes the vectors  $\theta$  and  $\omega$  as input

$$\dot{\theta}(\theta, \omega) = R\omega \quad (\text{C.19})$$

The Taylor approximation of the equation result in the following

$$\dot{\theta}(\theta, \omega) \approx \dot{\theta}(\bar{\theta}, \bar{\omega}) + \frac{\partial \dot{\theta}(\theta, \omega)}{\partial \theta} \Big|_{\substack{\omega=\bar{\omega} \\ \theta=\bar{\theta}}} \tilde{\theta} + \frac{\partial \dot{\theta}(\theta, \omega)}{\partial \omega} \Big|_{\substack{\omega=\bar{\omega} \\ \theta=\bar{\theta}}} \tilde{\omega} \quad (\text{C.20})$$

The equation is divided into 3 rows and the first row in the rotation matrix is

$$\dot{\theta}_1 = c(\theta_2) c(\theta_3) \omega_1 + c(\theta_2) s(\theta_3) \omega_2 - s(\theta_2) \omega_3 \quad (\text{C.21})$$

where  $s$  and  $c$  represents sinus and cosine respectively. The Euler rate  $\dot{\theta}_1$  is the first row in the Rotation matrix, linearised with respect to the Euler angles and angular velocities which yields

$$\begin{aligned}
 \dot{\theta}_1(\theta_1, \theta_2, \theta_3, \omega_1, \omega_2, \omega_3) = & \\
 c(\bar{\theta}_2) c(\bar{\theta}_3) \bar{\omega}_1 + c(\bar{\theta}_2) s(\bar{\theta}_3) \bar{\omega}_2 - s(\bar{\theta}_2) \bar{\omega}_3 & \\
 + (-s(\bar{\theta}_2) c(\bar{\theta}_3) \bar{\omega}_1 - s(\bar{\theta}_2) s(\bar{\theta}_3) \bar{\omega}_2 - c(\bar{\theta}_2) \bar{\omega}_3) \tilde{\theta}_2 & \\
 + (-s(\bar{\theta}_3) c(\bar{\theta}_2) \bar{\omega}_1 + c(\bar{\theta}_2) c(\bar{\theta}_3) \bar{\omega}_2) \tilde{\theta}_3 & \\
 + (c(\bar{\theta}_2) c(\bar{\theta}_3)) \tilde{\omega}_1 & \\
 + (c(\bar{\theta}_2) s(\bar{\theta}_3)) \tilde{\omega}_2 & \\
 - (s(\bar{\theta}_2)) \tilde{\omega}_3 &
 \end{aligned} \tag{C.22}$$

Second row in the rotation matrix

$$\begin{aligned}
 \dot{\theta}_2 = & \\
 (c(\theta_3) s(\theta_1) s(\theta_2) - c(\theta_1) s(\theta_3)) \omega_1 & \\
 + (c(\theta_1) c(\theta_3) + s(\theta_1) s(\theta_2) s(\theta_3)) \omega_2 & \\
 + (c(\theta_2) s(\theta_1)) \omega_3 &
 \end{aligned} \tag{C.23}$$

The linearised Euler rate  $\dot{\theta}_2$  is

$$\begin{aligned}
 \dot{\theta}_2(\theta_1, \theta_2, \theta_3, \omega_1, \omega_2, \omega_3) = & \\
 (c(\bar{\theta}_3) s(\bar{\theta}_1) s(\bar{\theta}_2) - c(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{\omega}_1 - (c(\bar{\theta}_1) c(\bar{\theta}_3) + s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{\omega}_2 + (c(\bar{\theta}_2) s(\bar{\theta}_1)) \bar{\omega}_3 & \\
 + ((c(\bar{\theta}_1) c(\bar{\theta}_3) s(\bar{\theta}_2) + s(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{\omega}_1 + (c(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) - s(\bar{\theta}_1) c(\bar{\theta}_3)) \bar{\omega}_2 + (c(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{\omega}_3) \tilde{\theta}_1 & \\
 + ((c(\bar{\theta}_3) s(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{\omega}_1 + (s(\bar{\theta}_1) c(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{\omega}_2 - (s(\bar{\theta}_1) s(\bar{\theta}_2)) \bar{\omega}_3) \tilde{\theta}_2 & \\
 + ((-s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) - c(\bar{\theta}_1) c(\bar{\theta}_3)) \bar{\omega}_1 + (s(\bar{\theta}_1) s(\bar{\theta}_2) c(\bar{\theta}_3) - c(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{\omega}_2) \tilde{\theta}_3 & \\
 + (c(\bar{\theta}_3) s(\bar{\theta}_1) s(\bar{\theta}_2) - s(\bar{\theta}_3) c(\bar{\theta}_1)) \tilde{\omega}_1 & \\
 + (s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) + c(\bar{\theta}_1) c(\bar{\theta}_3)) \tilde{\omega}_2 & \\
 + (s(\bar{\theta}_1) c(\bar{\theta}_2)) \tilde{\omega}_3 &
 \end{aligned} \tag{C.24}$$

Third row in the rotation matrix

$$\begin{aligned}
 \dot{\theta}_3 = & \\
 (s(\theta_1) s(\theta_3) + c(\theta_1) c(\theta_3) s(\theta_2)) \omega_1 & \\
 + (c(\theta_1) s(\theta_2) s(\theta_3) - s(\theta_1) c(\theta_3)) \omega_2 & \\
 + (c(\theta_1) c(\theta_2)) \omega_3 &
 \end{aligned} \tag{C.25}$$

The linearised Euler rate  $\dot{\theta}_3$  is

$$\begin{aligned}
\dot{\theta}_3(\theta_1, \theta_2, \theta_3, \omega_1, \omega_2, \omega_3) = & \\
(s(\bar{\theta}_1)s(\bar{\theta}_3) + c(\bar{\theta}_1)c(\bar{\theta}_3)s(\bar{\theta}_2))\bar{\omega}_1 + (c(\bar{\theta}_1)s(\bar{\theta}_2)s(\bar{\theta}_3) - c(\bar{\theta}_3)s(\bar{\theta}_1))\bar{\omega}_2 + (c(\bar{\theta}_1)c(\bar{\theta}_2))\bar{\omega}_3 & \\
+ ((c(\bar{\theta}_1)s(\bar{\theta}_3) - s(\bar{\theta}_1)s(\bar{\theta}_2)c(\bar{\theta}_3))\bar{\omega}_1 - (s(\bar{\theta}_1)s(\bar{\theta}_2)s(\bar{\theta}_3) - c(\bar{\theta}_1)c(\bar{\theta}_3))\bar{\omega}_2 - (s(\bar{\theta}_1)c(\bar{\theta}_2))\bar{\omega}_3)\tilde{\theta}_1 & \\
+ ((c(\bar{\theta}_1)c(\bar{\theta}_2)c(\bar{\theta}_3))\bar{\omega}_1 + (c(\bar{\theta}_1)c(\bar{\theta}_2)s(\bar{\theta}_3))\bar{\omega}_2 - (c(\bar{\theta}_1)s(\bar{\theta}_2))\bar{\omega}_3)\tilde{\theta}_2 & \\
+ ((s(\bar{\theta}_1)c(\bar{\theta}_3) - c(\bar{\theta}_1)s(\bar{\theta}_2)s(\bar{\theta}_3))\bar{\omega}_1 + (c(\bar{\theta}_1)s(\bar{\theta}_2)c(\bar{\theta}_3) + s(\bar{\theta}_1)s(\bar{\theta}_3))\bar{\omega}_2)\tilde{\theta}_3 & \\
+ (s(\bar{\theta}_1)s(\bar{\theta}_3) + c(\bar{\theta}_1)c(\bar{\theta}_3)s(\bar{\theta}_2))\tilde{\omega}_1 & \\
+ (s(\bar{\theta}_2)s(\bar{\theta}_3)c(\bar{\theta}_1) - s(\bar{\theta}_1)c(\bar{\theta}_3))\tilde{\omega}_2 & \\
+ (c(\bar{\theta}_1)c(\bar{\theta}_2))\tilde{\omega}_3 & \\
(C.26)
\end{aligned}$$

## Linear Velocity

The derivative of the position of the body in space is given by

$${}^I_B \dot{P} = {}^I_B \vec{v}_B = {}^I_B R^B \vec{v} \quad (C.27)$$

and takes  $\theta$  and  $v$  as input

$$\dot{P}(\theta, v) = Rv \quad (C.28)$$

First order Taylor approximation can be written as

$$\dot{P}(\theta, v) \approx \dot{P}(\bar{\theta}, \bar{v}) + \frac{\partial \dot{P}(\theta, v)}{\partial \theta} \Big|_{\substack{v=\bar{v} \\ \theta=\bar{\theta}}} \tilde{\theta} + \frac{\partial \dot{P}(\theta, v)}{\partial v} \Big|_{\substack{v=\bar{v} \\ \theta=\bar{\theta}}} \tilde{v} \quad (C.29)$$

The linearisation is also divided into 3 and the first row in the rotation matrix is

$$\dot{P}_1 = c(\theta_2)c(\theta_3)v_1 + c(\theta_2)s(\theta_3)v_2 - s(\theta_2)v_3 \quad (C.30)$$

The linearised velocity of the body  $\dot{P}_1$  is

$$\begin{aligned}
\dot{P}_1(\theta_1, \theta_2, \theta_3, v_1, v_2, v_3) = & \\
c(\bar{\theta}_2)c(\bar{\theta}_3)v_1 + c(\bar{\theta}_2)s(\bar{\theta}_3)v_2 - s(\bar{\theta}_2)v_3 & \\
+ (-s(\bar{\theta}_2)c(\bar{\theta}_3)\bar{v}_1 - s(\bar{\theta}_2)s(\bar{\theta}_3)\bar{v}_2 - c(\bar{\theta}_2)\bar{v}_3)\tilde{\theta}_2 & \\
+ (-s(\bar{\theta}_3)c(\bar{\theta}_2)\bar{v}_1 + c(\bar{\theta}_2)c(\bar{\theta}_3)\bar{v}_2)\tilde{\theta}_3 & \\
+ (c(\bar{\theta}_2)c(\bar{\theta}_3))\tilde{v}_1 & \\
+ (c(\bar{\theta}_2)s(\bar{\theta}_3))\tilde{v}_2 & \\
- (s(\bar{\theta}_2))\tilde{v}_3 & \\
(C.31)
\end{aligned}$$

Second row in the rotation matrix

$$\begin{aligned}
 \dot{P}_2 = & \\
 & (c(\theta_3) s(\theta_1) s(\theta_2) - c(\theta_1) s(\theta_3)) v_1 \\
 & + (c(\theta_1) c(\theta_3) + s(\theta_1) s(\theta_2) s(\theta_3)) v_2 \\
 & + (c(\theta_2) s(\theta_1)) v_3
 \end{aligned} \tag{C.32}$$

The linearised velocity of the body  $\dot{P}_2$  is

$$\begin{aligned}
 \dot{P}_2(\theta_1, \theta_2, \theta_3, v_1, v_2, v_3) = & \\
 & (c(\bar{\theta}_3) s(\bar{\theta}_1) s(\bar{\theta}_2) - c(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{v}_1 - (c(\bar{\theta}_1) c(\bar{\theta}_3) + s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{v}_2 + (c(\bar{\theta}_2) s(\bar{\theta}_1)) \bar{v}_3 \\
 & + ((c(\bar{\theta}_1) c(\bar{\theta}_3) s(\bar{\theta}_2) + s(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{v}_1 + (c(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) - s(\bar{\theta}_1) c(\bar{\theta}_3)) \bar{v}_2 + (c(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{v}_3) \tilde{\theta}_1 \\
 & + ((c(\bar{\theta}_3) s(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{v}_1 + (s(\bar{\theta}_1) c(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{v}_2 - (s(\bar{\theta}_1) s(\bar{\theta}_2)) \bar{v}_3) \tilde{\theta}_2 \\
 & + ((-s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) - c(\bar{\theta}_1) c(\bar{\theta}_3)) \bar{v}_1 + (s(\bar{\theta}_1) s(\bar{\theta}_2) c(\bar{\theta}_3) - c(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{v}_2) \tilde{\theta}_3 \\
 & + (c(\bar{\theta}_3) s(\bar{\theta}_1) s(\bar{\theta}_2) - s(\bar{\theta}_3) c(\bar{\theta}_1)) \tilde{v}_1 \\
 & + (s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) + c(\bar{\theta}_1) c(\bar{\theta}_3)) \tilde{v}_2 \\
 & + (s(\bar{\theta}_1) c(\bar{\theta}_2)) \tilde{v}_3
 \end{aligned} \tag{C.33}$$

Third differentiated row in the rotation matrix

$$\begin{aligned}
 \dot{P}_3 = & \\
 & (s(\theta_1) s(\theta_3) + c(\theta_1) c(\theta_3) s(\theta_2)) v_1 \\
 & + (-c(\theta_1) s(\theta_2) s(\theta_3) - s(\theta_1) c(\theta_3)) v_2 \\
 & + (-c(\theta_1) c(\theta_2)) v_3
 \end{aligned} \tag{C.34}$$

The linearised velocity of the body  $\dot{P}_3$  is

$$\begin{aligned}
 \dot{P}_3(\theta_1, \theta_2, \theta_3, v_1, v_2, v_3) = & \\
 & (s(\bar{\theta}_1) s(\bar{\theta}_3) + c(\bar{\theta}_1) c(\bar{\theta}_3) s(\bar{\theta}_2)) \bar{v}_1 + (c(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) - c(\bar{\theta}_3) s(\bar{\theta}_1)) \bar{v}_2 + (c(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{v}_3 \\
 & + ((c(\bar{\theta}_1) s(\bar{\theta}_3) - s(\bar{\theta}_1) s(\bar{\theta}_2) c(\bar{\theta}_3)) \bar{v}_1 - (s(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3) + c(\bar{\theta}_1) c(\bar{\theta}_3)) \bar{v}_2 - (s(\bar{\theta}_1) c(\bar{\theta}_2)) \bar{v}_3) \tilde{\theta}_1 \\
 & + ((c(\bar{\theta}_1) c(\bar{\theta}_2) c(\bar{\theta}_3)) \bar{v}_1 + (c(\bar{\theta}_1) c(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{v}_2 - (c(\bar{\theta}_1) s(\bar{\theta}_2)) \bar{v}_3) \tilde{\theta}_2 \\
 & + ((s(\bar{\theta}_1) c(\bar{\theta}_3) - c(\bar{\theta}_1) s(\bar{\theta}_2) s(\bar{\theta}_3)) \bar{v}_1 + (c(\bar{\theta}_1) s(\bar{\theta}_2) c(\bar{\theta}_3) + s(\bar{\theta}_1) s(\bar{\theta}_3)) \bar{v}_2) \tilde{\theta}_3 \\
 & + (s(\bar{\theta}_1) s(\bar{\theta}_3) + c(\bar{\theta}_1) c(\bar{\theta}_3) s(\bar{\theta}_2)) \tilde{v}_1 \\
 & + (s(\bar{\theta}_2) s(\bar{\theta}_3) c(\bar{\theta}_1) - s(\bar{\theta}_1) c(\bar{\theta}_3)) \tilde{v}_2 \\
 & + (c(\bar{\theta}_1) c(\bar{\theta}_2)) \tilde{v}_3
 \end{aligned} \tag{C.35}$$

## C.4 Linearisation of the Generated Forces

The equation for lift is derived in section 5.3 and shown in equation (5.16). That and gravity are the forces affecting the position of the X-Pro. This is illustrated in equation (6.1) in section 6.

$$F_{tot}(\theta, \Omega) = \sum_{i=1}^4 F_{Lift,i} - F_g \quad (\text{C.36})$$

Since the 2 forces are independent they will be linearised separately. First the gravity force will be linearised and takes  $\theta$  as input.

$$F_g(\theta) \approx F_g(\bar{\theta}) + \frac{\partial F_g(\theta)}{\partial \theta} \Big|_{\theta=\bar{\theta}} \tilde{\theta} \quad (\text{C.37})$$

The gravity force affecting the body is a function of the Euler angles in the rotation matrix. The Taylor approximation of the rotation matrix from the universal coordinate system to the body coordinate system is shown below, it is the transpose of the rotation matrix in equation (8.14) on page 45. These approximations depends on the mass and the gravity, the mass is a scalar and the gravity is a vector which only affects the z-axis.

$$\begin{bmatrix} F_g \end{bmatrix} = m \begin{bmatrix} {}^B R \\ I \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (\text{C.38})$$

First the taylor approximation of  $F_g$  on the x-axis

$$F_{g,x}(\theta_1, \theta_2, \theta_3) \approx \quad (\text{C.39})$$

$$mg(\sin(\bar{\theta}_1) \sin(\bar{\theta}_3) + \cos(\bar{\theta}_1) \sin(\bar{\theta}_2) \cos(\bar{\theta}_3)) \quad (\text{C.39})$$

$$+mg(\cos(\bar{\theta}_1) \sin(\bar{\theta}_3) - \sin(\bar{\theta}_1) \sin(\bar{\theta}_2) \cos(\bar{\theta}_3))\tilde{\theta}_1 \quad (\text{C.40})$$

$$+mg(\cos(\bar{\theta}_1) \cos(\bar{\theta}_2) \cos(\bar{\theta}_3))\tilde{\theta}_2 \quad (\text{C.41})$$

$$+mg(\sin(\bar{\theta}_1) \cos(\bar{\theta}_3) - \cos(\bar{\theta}_1) \sin(\bar{\theta}_2) \sin(\bar{\theta}_3))\tilde{\theta}_3 \quad (\text{C.42})$$

The taylor approximation of  $F_g$  on the y-axis

$$F_{g,y}(\theta_1, \theta_2, \theta_3) \approx \quad (\text{C.43})$$

$$mg(\cos(\bar{\theta}_1) \sin(\bar{\theta}_2) \sin(\bar{\theta}_3) - \sin(\bar{\theta}_1) \cos(\bar{\theta}_3)) \quad (\text{C.43})$$

$$+mg(-\sin(\bar{\theta}_1) \sin(\bar{\theta}_2) \sin(\bar{\theta}_3) - \cos(\bar{\theta}_1) \cos(\bar{\theta}_3))\tilde{\theta}_1 \quad (\text{C.44})$$

$$+mg(\cos(\bar{\theta}_1) \cos(\bar{\theta}_2) \sin(\bar{\theta}_3))\tilde{\theta}_2 \quad (\text{C.45})$$

$$+mg(\cos(\bar{\theta}_1) \sin(\bar{\theta}_2) \cos(\bar{\theta}_3) + \sin(\bar{\theta}_1) \sin(\bar{\theta}_3))\tilde{\theta}_3 \quad (\text{C.46})$$

And lastly the z-axis

$$F_{g,z}(\theta_1, \theta_2, \theta_3) \approx mg(\cos(\bar{\theta}_1) \cos(\bar{\theta}_2)) \quad (C.47)$$

$$+mg(-\sin(\bar{\theta}_1) \cos(\bar{\theta}_2))\tilde{\theta}_1 \quad (C.48)$$

$$+mg(-\cos(\bar{\theta}_1) \sin(\bar{\theta}_2))\tilde{\theta}_2 \quad (C.49)$$

That provides the taylor approximation of the gravity force.

The second part is the total lift force that takes  $\Omega$  as input.

$$F_{Lift}(\Omega) \approx F_{Lift}(\bar{\Omega}) + \frac{\partial F_{Lift}(\Omega)}{\partial \Omega} \Big|_{\Omega=\bar{\Omega}} \tilde{\Omega} \quad (C.50)$$

The values for the lift force is found in section 5.3 to

$$F_{Lift}(\Omega) = 0.226 \cdot 10^{-3} \cdot \Omega^2 - 2.49 \cdot 10^{-3} \cdot \Omega + 11.97 \cdot 10^{-3} \quad [N] \quad (C.51)$$

The inputs for the lift force is the angular velocity on each of the rotors. Since neither rotor provides lift on other axis than the z-axis and all rotors are modelled the same way, there will be a similar result for all rotors. For the i'th rotor, the Taylor approximation on the z-axis gives

$$\begin{aligned} F_{Lift,i}(\Omega_i) &= \\ 0.226 \cdot 10^{-3} \cdot \bar{\Omega}_i^2 &- 2.49 \cdot 10^{-3} \cdot \bar{\Omega}_i + 11.97 \cdot 10^{-3} \\ +(0.452 \cdot 10^{-3} \cdot \bar{\Omega}_i - 2.49 \cdot 10^{-3})\tilde{\Omega}_i & \end{aligned} \quad (C.52)$$

## C.5 Linearisation of the Generated Torque

The torques affecting the X-Pro are generated by the lift force and the drag force.

$$\tau = \tau_{Lift} + \tau_{Drag} \quad (C.53)$$

$$= \sum_{i=1}^4 F_{Lift,i} \times P_i + \sum_{i=1}^4 F_{Drag,i} \times P_i \quad (C.54)$$

Substituting the cross product with the skew symmetric matrix gives a complete model for the generated torque

$$\begin{bmatrix} \tau \end{bmatrix} = \begin{bmatrix} -S(P_1)F_{L,1} - S(P_1)F_{D,1} & \dots & \dots & -S(P_4)F_{L,4} - S(P_4)F_{D,4} \end{bmatrix} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (\text{C.55})$$

Linearising  $F_{Lift}$  is done in the previous section, which means  $F_{Drag}$  is missing. This is done similar to that of  $F_{Lift}$  using Taylor approximations.

$$F_{drag}(\Omega) \approx F_{drag}(\bar{\Omega}) + \frac{\partial F_{drag}(\Omega)}{\partial \Omega} \Big|_{\Omega=\bar{\Omega}} \tilde{\Omega} \quad (\text{C.56})$$

The equation for the drag force is calculated in equation (5.17) in section 5.3. The resulting Taylor approximation for the i'th rotor yields

$$\begin{aligned} F_{drag}(\Omega) = & \\ 26.6 \cdot 10^{-6} \cdot \bar{\Omega}_i^2 - 1.93 \cdot 10^{-6} \cdot \bar{\Omega}_i + 18.8 \cdot 10^{-3} \\ + (53.2 \cdot 10^{-6} \cdot \bar{\Omega}_i - 1.93 \cdot 10^{-6}) \tilde{\Omega}_i \end{aligned} \quad (\text{C.57})$$

The non-linear equations describing the model of the X-Pro has now been linearised and can be used for the state space representation of the system.

## C.6 State Space Representation of the Complete Model

The linearised equations will now be used to represent the complete model by the use of state space representation. The representation will start with the body, then rotor, motor and finally mixer. In this way the state space representation is able to represent the output as a function of the input. This representation will be done symbolic, which leaves reduction to be done in the controller chapter.

### C.6.1 State Space model of the Body

The state vector was first presented in chapter 9. The state space representation of the body will have index 1 and be as follow.

$$\dot{x}_1 = \mathbf{A}_1 x_1 + \mathbf{B}_1 u_1 \quad (\text{C.58})$$

$$y_1 = \mathbf{C}_1 x_1 + \mathbf{D}_1 u_1 \quad (\text{C.59})$$

The dimensions of the body state space representation are as follow.

$$\begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} = \begin{bmatrix} \mathbf{A}_1 \end{bmatrix}_{12 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \begin{bmatrix} \mathbf{B}_1 \end{bmatrix}_{12 \times 6} \begin{bmatrix} F \\ \tau \end{bmatrix}_{6 \times 1} \quad (\text{C.60})$$

$$\begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} = \begin{bmatrix} \mathbf{C}_1 \end{bmatrix}_{12 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \begin{bmatrix} \mathbf{D}_1 \end{bmatrix}_{12 \times 6} \begin{bmatrix} F \\ \tau \end{bmatrix}_{6 \times 1} \quad (\text{C.61})$$

The  $\mathbf{C}_1$  matrix will be and identity matrix to map the states to the output and as the output not directly depends on the input,  $\mathbf{D}_1$  will be zero.

### C.6.2 Rotor State Space Representation

To be able to include the rotor state space representation in the complete model, the body and rotor model must be combined as they both partly depend on the same parameter. The state space representation for the rotor will be marked index 2.

$$\dot{x}_2 = \mathbf{A}_2 x_2 + \mathbf{B}_2 u_2 \quad (\text{C.62})$$

$$y_2 = \mathbf{C}_2 x_2 + \mathbf{D}_2 u_2 \quad (\text{C.63})$$

As the rotor uses the same states as for the body,  $x_2 = x_1$  and the representation will be

$$\begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} = \begin{bmatrix} \mathbf{A}_2 \end{bmatrix}_{12 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \begin{bmatrix} \mathbf{B}_2 \end{bmatrix}_{6 \times 4} \begin{bmatrix} \Omega \end{bmatrix}_{4 \times 1} \quad (\text{C.64})$$

$$\begin{bmatrix} F \\ \tau \end{bmatrix}_{6 \times 1} = \begin{bmatrix} \mathbf{C}_2 \end{bmatrix}_{6 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \begin{bmatrix} \mathbf{D}_2 \end{bmatrix}_{6 \times 4} \begin{bmatrix} \Omega \end{bmatrix}_{4 \times 1} \quad (\text{C.65})$$

The equations for  $F$  and  $\tau$  only depends on  $\theta$  and  $\Omega$ , as can be seen in equations (6.1) and (6.6). The  $\mathbf{A}_2$  and  $\mathbf{B}_2$  will be zero and  $y_2$  only depends on  $\mathbf{C}_2$  and  $\mathbf{D}_2$ .

As  $y_2$  is the input in equation (C.60) and (C.61) a substitution results in

$$u_1 = y_2 = \mathbf{C}_2 x_1 + \mathbf{D}_2 u_2 \quad (\text{C.66})$$

$$\dot{x}_1 = \mathbf{A}_1 x_1 + \mathbf{B}_1 (\mathbf{C}_2 x_1 + \mathbf{D}_2 u_2) \quad (C.67)$$

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2) x_1 + \mathbf{B}_1 \mathbf{D}_2 u_2 \quad (C.68)$$

$$y_1 = \mathbf{C}_1 x_1 + \mathbf{D}_1 (\mathbf{C}_2 x_1 + \mathbf{D}_2 u_2) \quad (C.69)$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2) x_1 + \mathbf{D}_1 \mathbf{D}_2 u_2 \quad (C.70)$$

$\mathbf{C}_1$  is an  $12 \times 12$  identity matrix and  $\mathbf{D}_1$  is zero. The combined state space representation for the body and rotor is

$$\begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} = \left[ \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2 \right]_{12 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \left[ \mathbf{B}_1 \mathbf{D}_2 \right]_{12 \times 4} \begin{bmatrix} \Omega \end{bmatrix}_{4 \times 1} \quad (C.71)$$

$$\begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} = \left[ \mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2 \right]_{12 \times 12} \begin{bmatrix} P \\ v \\ \theta \\ \omega \end{bmatrix}_{12 \times 1} + \left[ \mathbf{D}_1 \mathbf{D}_2 \right]_{12 \times 4} \begin{bmatrix} \Omega \end{bmatrix}_{4 \times 1} \quad (C.72)$$

A merged state space representation of the body and rotor has been introduced. Next the state space representation will be extended with motor as input.

### C.6.3 Motor + Rotor + Body State Space

General state space representation of the motor model

$$\dot{x}_3 = \mathbf{A}_3 x_3 + \mathbf{B}_3 u_3 \quad (C.73)$$

$$y_3 = \mathbf{C}_3 x_3 + \mathbf{D}_3 u_3 \quad (C.74)$$

The representation includes 4 motors, each with one input and one output. The dimensions of the motor state vector will be extended as the models of the motors are estimated as 2. order polynomials.

$$\begin{bmatrix} \dot{\Omega} \end{bmatrix}_{8 \times 1} = \left[ \mathbf{A}_3 \right]_{8 \times 8} \begin{bmatrix} \Omega \end{bmatrix}_{8 \times 1} + \left[ \mathbf{B}_3 \right]_{8 \times 4} \begin{bmatrix} \text{PWM} \end{bmatrix}_{4 \times 1} \quad (C.75)$$

$$\begin{bmatrix} \Omega \end{bmatrix}_{4 \times 1} = \left[ \mathbf{C}_3 \right]_{4 \times 8} \begin{bmatrix} \Omega \end{bmatrix}_{8 \times 1} + \left[ \mathbf{D}_3 \right]_{4 \times 4} \begin{bmatrix} \text{PWM} \end{bmatrix}_{4 \times 1} \quad (C.76)$$

To merge the models, the output from the motor,  $y_3$  will be the input,  $u_2$  for the body/rotor representation.

$$u_2 = y_3 = \mathbf{C}_3 x_3 + \mathbf{D}_3 u_3 \quad (C.77)$$

The state space representation then results in

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2)x_1 + \mathbf{B}_1 \mathbf{D}_2 u_2 \quad (C.78)$$

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2)x_1 + \mathbf{B}_1 \mathbf{D}_2(\mathbf{C}_3 x_3 + \mathbf{D}_3 u_3) \quad (C.79)$$

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2)x_1 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 u_3 \quad (C.80)$$

$$\dot{x}_3 = \mathbf{A}_3 x_3 + \mathbf{B}_3 u_3 \quad (C.81)$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2)x_1 + \mathbf{D}_1 \mathbf{D}_2 u_2 \quad (C.82)$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2)x_1 + \mathbf{D}_1 \mathbf{D}_2(\mathbf{C}_3 x_3 + \mathbf{D}_3 u_3) \quad (C.83)$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2)x_1 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 u_3 \quad (C.84)$$

$$y_3 = \mathbf{C}_3 x_3 + \mathbf{D}_3 u_3 \quad (C.85)$$

After the substitution the state space representation can be based on equation (C.80), (C.81), (C.84) and (C.85) and can be represented as a function of the PWM input.

$$\begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \\ \Omega \end{bmatrix}_{20 \times 1} = \begin{bmatrix} \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2 & \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 \\ 0 & \mathbf{A}_3 \end{bmatrix}_{20 \times 20} \begin{bmatrix} P \\ v \\ \theta \\ \omega \\ \Omega \end{bmatrix}_{20 \times 1} + \begin{bmatrix} \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 \\ \mathbf{B}_3 \end{bmatrix}_{20 \times 4} \begin{bmatrix} \text{PWM} \end{bmatrix}_{4 \times 1} \quad (C.86)$$

$$\begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \\ \Omega \end{bmatrix}_{16 \times 1} = \begin{bmatrix} \mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2 & \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 \\ 0 & \mathbf{C}_3 \end{bmatrix}_{16 \times 20} \begin{bmatrix} P \\ v \\ \theta \\ \omega \\ \Omega \end{bmatrix}_{20 \times 1} + \begin{bmatrix} \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \\ \mathbf{D}_3 \end{bmatrix}_{16 \times 4} \begin{bmatrix} \text{PWM} \end{bmatrix}_{4 \times 1} \quad (C.87)$$

After the extension of the motor, the final model, mixer, will be added to complete the state space representation.

#### C.6.4 Mixer + Motor + Rotor + Body State Space

General state space representation of the mixer model

$$\dot{x}_4 = \mathbf{A}_4 x_4 + \mathbf{B}_4 u_4 \quad (C.88)$$

$$y_4 = \mathbf{C}_4 x_4 + \mathbf{D}_4 u_4 \quad (C.89)$$

The input for the mixer are the control signals, throttle, yaw, pitch and roll. The output is the PWM signal which are the input signals for the motors. The state space representation of the

mixer can then be written as

$$\left[ \begin{array}{c} \cdot \\ \text{PMW} \end{array} \right]_{4 \times 1} = \left[ \begin{array}{c} \mathbf{A}_4 \end{array} \right]_{4 \times 4} \left[ \begin{array}{c} \text{PWM} \end{array} \right]_{4 \times 1} + \left[ \begin{array}{c} \mathbf{B}_4 \end{array} \right]_{4 \times 4} \left[ \begin{array}{c} \text{Throttle} \\ \text{Yaw} \\ \text{Pitch} \\ \text{Roll} \end{array} \right]_{4 \times 1} \quad (\text{C.90})$$

$$\left[ \begin{array}{c} \cdot \\ \text{PWM} \end{array} \right]_{4 \times 1} = \left[ \begin{array}{c} \mathbf{C}_4 \end{array} \right]_{4 \times 4} \left[ \begin{array}{c} \text{PWM} \end{array} \right]_{4 \times 1} + \left[ \begin{array}{c} \mathbf{D}_4 \end{array} \right]_{4 \times 4} \left[ \begin{array}{c} \text{Throttle} \\ \text{Yaw} \\ \text{Pitch} \\ \text{Roll} \end{array} \right]_{4 \times 1} \quad (\text{C.91})$$

As the output of the mixer is the input to the motor,  $y_4$  will be substituted into  $u_3$ .

$$u_3 = y_4 = \mathbf{C}_4 x_4 + \mathbf{D}_4 u_4 \quad (\text{C.92})$$

The state space representation then results in

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2) x_1 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 u_3 \quad (\text{C.93})$$

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2) x_1 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 (\mathbf{C}_4 x_4 + \mathbf{D}_4 u_4) \quad (\text{C.94})$$

$$\dot{x}_1 = (\mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2) x_1 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{C}_4 x_4 + \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 u_4 \quad (\text{C.95})$$

$$\dot{x}_3 = \mathbf{A}_3 x_3 + \mathbf{B}_3 u_3 \quad (\text{C.96})$$

$$\dot{x}_3 = \mathbf{A}_3 x_3 + \mathbf{B}_3 (\mathbf{C}_4 x_4 + \mathbf{D}_4 u_4) \quad (\text{C.97})$$

$$\dot{x}_3 = \mathbf{A}_3 x_3 + \mathbf{B}_3 \mathbf{C}_4 x_4 + \mathbf{B}_3 \mathbf{D}_4 u_4 \quad (\text{C.98})$$

$$\dot{x}_4 = \mathbf{A}_4 x_4 + \mathbf{B}_4 u_4 \quad (\text{C.99})$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2) x_1 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 u_3 \quad (\text{C.100})$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2) x_1 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 (\mathbf{C}_4 x_4 + \mathbf{D}_4 u_4) \quad (\text{C.101})$$

$$y_1 = (\mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2) x_1 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 x_3 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{C}_4 x_4 + \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 u_4 \quad (\text{C.102})$$

$$y_3 = \mathbf{C}_3 x_3 + \mathbf{D}_3 (\mathbf{C}_4 x_4 + \mathbf{D}_4 u_4) \quad (\text{C.103})$$

$$y_3 = \mathbf{C}_3 x_3 + \mathbf{D}_3 \mathbf{C}_4 x_4 + \mathbf{D}_3 \mathbf{D}_4 u_4 \quad (\text{C.104})$$

$$y_4 = \mathbf{C}_4 x_4 + \mathbf{D}_4 u_4 \quad (\text{C.105})$$

The equations can then be presented by matrices in state space representation.

$$\begin{aligned}
 \begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \\ \Omega \\ \text{PWM} \end{bmatrix}_{24 \times 1} &= \begin{bmatrix} \mathbf{A}_1 + \mathbf{B}_1 \mathbf{C}_2 & \mathbf{B}_1 \mathbf{D}_2 \mathbf{C}_3 & \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{C}_4 \\ 0 & \mathbf{A}_3 & \mathbf{B}_3 \mathbf{C}_4 \\ 0 & 0 & \mathbf{A}_4 \\ \Omega & \Omega & \Omega \\ \text{PWM} & \text{PWM} & \text{PWM} \end{bmatrix}_{24 \times 24}^{24 \times 1} \\
 &\quad + \begin{bmatrix} P \\ v \\ \theta \\ \omega \\ \Omega \\ \text{PWM} \end{bmatrix}_{24 \times 1}^{24 \times 1} \\
 &\quad + \begin{bmatrix} \mathbf{B}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \\ \mathbf{B}_3 \mathbf{D}_4 \\ \mathbf{B}_4 \end{bmatrix}_{24 \times 4}^{24 \times 1} \\
 &\quad \left[ \begin{array}{c} \text{Throttle} \\ \text{Yaw} \\ \text{Pitch} \\ \text{Roll} \end{array} \right]_{4 \times 1}^{4 \times 1} \tag{C.106}
 \end{aligned}$$
  

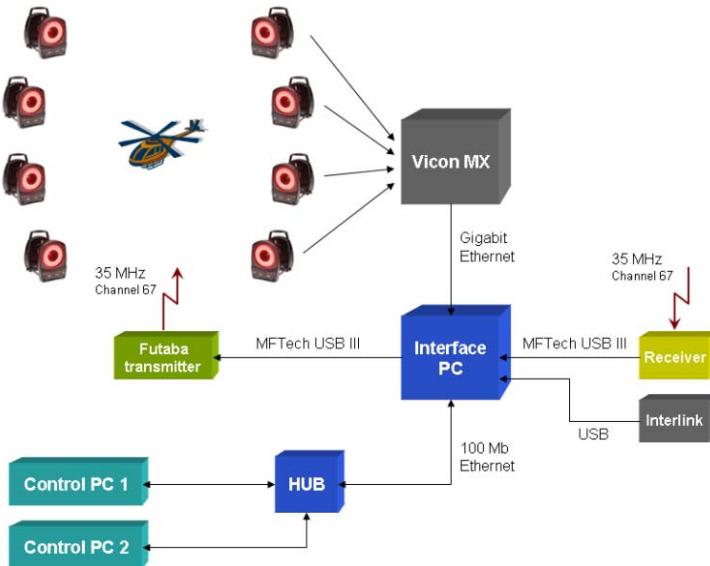
$$\begin{aligned}
 \begin{bmatrix} \dot{P} \\ v \\ \theta \\ \omega \\ \Omega \\ \text{PWM} \end{bmatrix}_{20 \times 1} &= \begin{bmatrix} \mathbf{C}_1 + \mathbf{D}_1 \mathbf{C}_2 & \mathbf{D}_1 \mathbf{D}_2 \mathbf{C}_3 & \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{C}_4 \\ 0 & \mathbf{C}_3 & \mathbf{D}_3 \mathbf{C}_4 \\ 0 & 0 & \mathbf{C}_4 \\ \Omega & \Omega & \Omega \\ \text{PWM} & \text{PWM} & \text{PWM} \end{bmatrix}_{20 \times 24}^{20 \times 1} \\
 &\quad + \begin{bmatrix} P \\ v \\ \theta \\ \omega \\ \Omega \\ \text{PWM} \end{bmatrix}_{24 \times 1}^{24 \times 1} \\
 &\quad + \begin{bmatrix} \mathbf{D}_1 \mathbf{D}_2 \mathbf{D}_3 \mathbf{D}_4 \\ \mathbf{D}_3 \mathbf{D}_4 \\ \mathbf{D}_4 \end{bmatrix}_{20 \times 4}^{20 \times 1} \\
 &\quad \left[ \begin{array}{c} \text{Throttle} \\ \text{Yaw} \\ \text{Pitch} \\ \text{Roll} \end{array} \right]_{4 \times 1}^{4 \times 1} \tag{C.107}
 \end{aligned}$$



# Appendix D

## Motion Tracking Laboratory

This appendix describes the Motion Tracking LABoratory (MTLAB), which is used in order to investigate the position and angles of the X-Pro. MTLAB features a Vicon MX Motion Capture System with 8 cameras [BlCH08a], which covers an indoor laboratory. The cameras are able to measure the position of the center of special markers inside the room and send the data to a PC.



**Figure D.1:** MTLAB general setup. The figure illustrates the routing between the elements of the setup [BlCH08a].

The cameras are of type MX3+ and are able to keep the capture speed at 100 frames pr. second, the maximum detected markers are set to be 4. A camera is shown in figure D.2.

The cameras has strobes that emits flashing light between 50 and 2000 times a second [Vic08]. With a resolution of  $659 \times 493$ , each pixel in a camera is light sensitive and measures the amount of light reflected by the markers that hits it, during a given period of time.

The data from the cameras is sent from the Vicon MX through Gigabit Ethernet to an Interface



**Figure D.2:** MX3+ camera in laboratory.

PC with Vicon IQ software installed, illustrated in figure D.1. The Vicon IQ software calculates the position and Euler angles of the measured object based on a model defined in the Vicon IQ software. This model is created using the spheres mounted on the object seen by the infrared cameras resulting in a software representation of the object. The center of mass of the object then needs to be defined in the model. Based on this reference point, the system calculates the angles and position. These data can be transferred over a 100 Mbit ethernet via a hub to any control PC. The data is used for further calculations in order to find e.g. velocity or acceleration of a moving object.

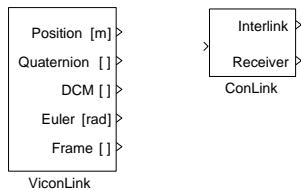
Furthermore the figure shows 3 blocks connected to the Vicon Interface PC. These are

1. Interlink - receives signals from USB R/C
2. Receiver - receives R/C transmitter signals
3. Transmitter - transmits signals from the Interface PC

# Appendix E

## Available Simulink Blocks

The MTLAB offers a variety of Simulink blocks which can be used, most notably the ViconLink and ConLink blocks, shown in figure E.1, which will be described in the following.



**Figure E.1:** The ViconLink and ConLink Simulink blocks.

### E.1 ViconLink

The ViconLink block establishes a connection to the Interface PC from the control PC as seen in figure D.1 on page 121. Then it collects the Euler angles and the position calculated by the Vicon system and makes this data available for Simulink. Quaternion, DCM and Frame is not used in control of the X-Pro.

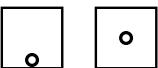
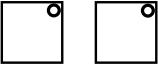
### E.2 ConLink

The ConLink block [B1CH08b] in Simulink is responsible for sending and receiving control signals, through the USB interfaces between the Controller PC and the Interface PC. The diagram of how the Controller PC and Interface PC is connected, is sketched in figure D.1. The ConLink block is used for 2 purposes.

1. To send RF signals from the Controller PC to the X-Pro (block input)
2. To receive RF signals on the Controller PC, sent from to the X-Pro (block receiver output)

Furthermore it is possible to use an USB R/C transmitter, where the ConLink interlink output is used. The Interface PC is connected to the R/C transmitter through a cable, and the output on the R/C transmitter can then be controlled by flipping the trainer switch and send a value to the Interface PC between -100 to 100 through the ConLink Simulink module. The range of 200 steps are then mapped by the Interface PC to make the R/C transmitter send with a 1 ms pulse period for -100, 1,5 ms for 0 ms and 2 ms for 100.

To understand the way the transmitted signals changes compared to the joystick manipulations, tests are performed. The joysticks are set in three positions and the received signals in the Conlink Simulink block on the Controller PC are measured. The result from this test can be seen in table E.1.

Controller position	Throttle	Yaw	Pitch	Roll
	-47.0000	-26.8276	1.0091	-20.7590
	48.7070	48.4565	-78.0000	59.7629
	-49.0000	-80.0000	83.9909	-98.3478

**Table E.1:** To the left three different positions of the R/C controller sticks, with the corresponding received R/C signal to the right.

From these two tests, 4 functions can be derived in order to map the received R/C signal to what is sent from ConLink and vice versa. The four functions are shown below.

$$\text{Throttle} = 0.97x - 48.8$$

$$\text{Yaw} = 0.59x - 23.8$$

$$\text{Pitch} = -0.58x - 33.1$$

$$\text{Roll} = -0.55x - 6.6$$

Appendix **F**

## Calculating the Inertia tensor

We need to calculate the inertia tensor of the whole body in the center of mass. The inertia tensor are defined as the following [Cra89, p. 191].

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (\text{F.1})$$

To find the inertia in each direction, the mass moment and mass product of inertia needs to be calculated from the following equations.

$$I_{xx} = \iiint_v (y^2 + z^2) \rho dv \quad (\text{F.2})$$

$$I_{yy} = \iiint_v (x^2 + z^2) \rho dv \quad (\text{F.3})$$

$$I_{zz} = \iiint_v (x^2 + y^2) \rho dv \quad (\text{F.4})$$

$$I_{xy} = \iiint_v xy \rho dv \quad (\text{F.5})$$

$$I_{xz} = \iiint_v xz \rho dv \quad (\text{F.6})$$

$$I_{yz} = \iiint_v yz \rho dv \quad (\text{F.7})$$

Where  $\rho$  is defined as the density of a body, which is a ratio of its mass  $m$  to its volume  $V$ :

$$\rho = \frac{m}{V} \quad (\text{F.8})$$

The strategy for calculating the inertia tensor for the whole body can be done in six steps.

1. split the body into different simple geometric objects.
2. chose a frame  $\{A\}$  for each of the objects. The frames origin should be placed in the centre of mass for the object, and the primary axis aligned with the symmetry axes, to simplify the inertia tensor for the object.
3. Calculate the inertia tensor for each object using the mentioned equation for the mass moment and mass product of inertia.
4. Rotate the inertia tensor according to the difference between the objects frame  $\{A\}$  and the universal frame to get a new inertia tensor for the rotated frame

$$\{A^*\} ({}^{A^*}I = {}_A^*R \ {}^A I \ {}_A^*R^T)$$

5. Use the parallel axis theorem [Cra89, p. 193] to move the inertia tensor of the object according to the difference in the body's center of mass and the objects center of mass.
6. When the inertia tensor for all the objects have been found they can all be summed together to get the inertia tensor for the entire body.

The inertia tensor are calculated by the use of a Matlab script developed by project group 06gr831 [06g06]. The file are located on the cd, CD\m-files\inertia-tensor\dynamics\_data.m. The inertia tensor are calculated by the m-file to be

$$I = \begin{bmatrix} 0.14999 & 0.00013 & 0 \\ 0.00013 & 0.14999 & 0 \\ 0 & 0 & 0.28190 \end{bmatrix} \quad (\text{F.9})$$

It is assumed that the values away from the diagonal in the I matrix are small enough to be disregarded, when the shape of the X-Pro are taken into account. Therefore, the inertia tensor used in the project will be the following

$$I = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.28 \end{bmatrix} \quad (\text{F.10})$$