Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Modelling and control of an advanced camera gimbal

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Jakob Johansson

LiTH-ISY-EX--12/4644--SE

Linköping 2012

Linköpings universitet
TEKNISKA HÖGSKOLAN

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

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Modelling and control of an advanced camera gimbal

Examensarbete utfört i Reglerteknik
vid Tekniska högskolan vid Linköpings universitet
av

Jakob Johansson

LiTH-ISY-EX--12/4644--SE

Handledare: **Mårten Svanfeldt**
Intuitive Aerial
Manon Kok
ISY, Linköpings universitet

Examinator: **David Törnqvist**
ISY, Linköpings universitet

Linköping, 27 november 2012

Avdelning, Institution
Division, Department

Institutionen för systemteknik
Department of Electrical Engineering
SE-581 83 Linköping

Datum
Date

2012-11-27

Språk
Language

□ Svenska/Swedish

☒ Engelska/English

□ ________________

Rapporttyp
Report category

□ Licentiatavhandling

☒ Examensarbete

□ C-uppsats

□ D-uppsats

□ Övrig rapport

□ ________________

ISBN

—

ISRN

LiTH-ISY-EX--12/4644--SE

Serietitel och serienummer
Title of series, numbering

ISSN

—

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-85952

Titel
Title

Modellering och styrning av en avancerad kameragimbal

Modelling and control of an advanced camera gimbal

Författare
Author

Jakob Johansson

Sammanfattning
Abstract

This thesis is about the modelling and control of three axis camera pan-roll-tilt unit (gimbal) which was meant to be attached to a multi rotor platform for aerial photography. The goal of the thesis was to develop a control structure for steering and active gyro stabilization of the gimbal, with aid from a mathematical model of the gimbal. Lagrange equations, together with kinematic equations and data from CAD drawings, were used to calculate a dynamics model of the gimbal. This model was set up as a Simulink simulation environment. Code for sensor reading and actuator control was written to the gimbal's microprocessor and the code for the control structure in the gimbal was developed in parallel with a control structure in the simulation environment. The thesis resulted in a method for mathematical modelling of the gimbal and a control structure, for steering and active gyro stabilization of the gimbal, implemented in its control unit as well as in the simulation environment.

Nyckelord
Keywords

gimbal, control, modelling, kinematics, dynamics

Abstract

This thesis is about the modelling and control of three axis camera pan-roll-tilt unit (gimbal) which was meant to be attached to a multi rotor platform for aerial photography. The goal of the thesis was to develop a control structure for steering and active gyro stabilization of the gimbal, with aid from a mathematical model of the gimbal. Lagrange equations, together with kinematic equations and data from CAD drawings, were used to calculate a dynamics model of the gimbal. This model was set up as a Simulink simulation environment. Code for sensor reading and actuator control was written to the gimbal's microprocessor and the code for the control structure in the gimbal was developed in parallel with a control structure in the simulation environment. The thesis resulted in a method for mathematical modelling of the gimbal and a control structure, for steering and active gyro stabilization of the gimbal, implemented in its control unit as well as in the simulation environment.

Sammanfattning

Detta examensarbete handlar om modellering och styrning av av en 3-axlig s.k. pan-roll-tilt-enhet (gimbal) till en kamera för flygfotografering från en multikopter. Målet med examensarbetet var att utveckla en struktur för styrning och aktiv gyrostabilisering av gimbalen med hjälp av en matematisk modell av den. Lagrangeekvationer, tillsammans med kinematiska ekvationer och data från CAD-ritningar, användes till att räkna fram en modell av gimbalens dynamik. Denna modell sattes upp som en simuleringsmiljö i Simulink. Programmeringskod för sensoravläsning och aktuatorstyrning skrevs till gimbalens egna mikroprocessor och koden för styrstrukturen till gimbalen utveckades parallellt med en styrstruktur i simuleringsmiljön. Examensarbetet resulterade i en metod för matematisk modellering av en gimbal och en struktur för styrning och aktiv gyrostabilisering implementerad i gimbalens styrenhet samt i en simuleringsmiljö.

Acknowledgments

First of all, I would like to thank Intuitive Aerial for the opportunity to do this thesis and the valuable experience of a young company in a very exciting phase. My thanks goes to my supervisor Manon Kok and examiner David Törnqvist at ISY, for their great patience valuable input on the report, and Per Skoglar at ISY, for his input early in the thesis. Last by not least I would like to thank my parents and the rest of the family for their everlasting love and support.

Linköping, November 2012
Jakob Johansson

Contents

Notation

Operators and functions

Notation	Meaning
$\dot{\mathbf{X}}$	The time derivative of **X**
$\mathbf{1}_n$	An identity matrix of size n.
$\mathbf{I}$	Mass moment of inertia
${}^{B}\mathbf{r}_{P}$	The coordinate *P* expressed in coordinate frame *B*.
$\tilde{\mathbf{r}}_i$	The skew-symmetric matrix form of vector $\mathbf{r}_i$
${}^{G}\mathcal{T}_{B}$	Rotational (**R**), translational (**d**) or homogeneous (**T**) transformation from coordinate frame *B* to frame *G*.
$\mathcal{T}_i^T$	Transpose of matrix $\mathcal{T}_i$
Θ	Joint angular position vector.
θ_i	Component *i* of vector Θ (joint parameter).
${}_{G}^{B}\Omega_{B}$	Joint angular velocity vector of coordinate frame *B* with respect to frame *G*, expressed in frame *B*.
ω_i	Component *i* of vector Ω.
α	Attitude of the IMU relative the ground.
Φ	Angular velocity of the IMU around its own axes.
ϕ_i	Component *i* of vector Φ.
Γ	Cartesian acceleration of the IMU.

Abbreviations

Abbreviation	Meaning
CAD	Computer Aided Design
IA	Intuitive Aerial
DOF	Degrees of Freedom
IMU	Inertial Measurement Unit
MBS	Multi body System
UAV	Unmanned Aerial Vehicle
CM	Center of mass
PWM	Pulse-width modulation
rad	radians

1

Introduction

This chapter covers the background of this thesis as well as its original purposes, goals and limitations. A survey of earlier work in scientific literature is also included in this chapter.

1.1 General introduction

1.1.1 Background

Intuitive Aerial AB, henceforth referred to as IA, develops a UAV system for aerial photography comprising a multi-rotor aerial vehicle, ground station and a remote control system. The vehicle, hereafter referred to as the *platform*, is made out of carbon fiber composites and ABS plastics and is controlled by a built in electronic control system that keeps the platform balanced as well as on course to an intended target. An important part of the UAV system, and the basis for this thesis, is a three axis camera mount, henceforth called the *gimbal* attached to the platform.

1.1.2 The camera gimbal

The basis for this thesis is a prototype three axis (yaw, roll, pitch) camera gimbal designed by IA parallel to the thesis work for mounting of different types of cameras. Each of the three axes is independently driven by motors and gears with encoder feedback which gives angular position and velocity feedback. The gimbal is also equipped with an Inertial Measurement Unit (IMU), consisting of gyroscopes and accelerometers and is placed at the camera mounting point.

1.1.3 Outline of the thesis

This thesis is divided into four chapters:

This chapter (Chapter 1) is the introduction part where the original purposes and goals (section 1.2) are listed together with the later changes to these goals that were made during the thesis work. The chapter also contains a short survey of the earlier work within the field (section 1.3).

Chapter 2 is about the mathematic Modelling of the gimbal where the mechanical, sensory and input properties (section 2.1) of the gimbal are presented. Section 2.2 is about the kinematic equations used in the dynamics as well as in the control structure of the gimbal. Section 2.3 is about how Lagrange equations and actuator dynamics are used to calculate the mathematical model of the gimbal. Section 2.4 is how the unknown friction parameters in the gimbal joints and sensory noise are identified using measurements from experiments.

Chapter 3 is about the development of control structures in the simulation environment (section 3.1) and in the gimbal control unit (section 3.2).

Chapter 4 contains the conclusions drawn from the thesis, what could be done differently and suggestions for future work.

1.2 Purpose of the thesis

This thesis aims to design, implement and evaluate a stabilization and control system for a three axis camera gimbal attached to a multi-rotor helicopter. The control system should compute an optimal set of control outputs to aim the camera under given constraints such as maximal possible velocities, accelerations, angles etc. The control system should also keep the camera stable in reference to the ground. The system design must also consider implementation limitations including processing power requirements and limitations in accuracy of available sensor data.

To aid the control design, a model of the gimbal should be developed based on mechanical equations and experiments.

1.2.1 Goals

The thesis aims to accomplish the following goals:

Model

A model of the gimbal based on dynamic equations and experimental results. The model is meant to be presented in a Simulink model that can be integrated with the already by IA developed platform models for use in later work.

Gyro stabilization

One goal is to develop a system for compensation of small motional disturbances on the gimbal using active control. One possible definition is that the gimbal should be stable enough to aim the camera within a small angular tolerance while it is disturbed by a motion of a certain frequency. The goal is that the gimbal will be stable enough for a camera to still take good pictures.

Steering

The gimbal should be able to switch targets as quickly and smoothly as possible.

Dynamic targeting

The gimbal should be able to aim at a dynamic point, e.g. a continuous function of coordinates, given motion constraints such as maximal possible velocities, accelerations, angles etc.

Other challenges

- The control system should be robust enough to cope with various types of cameras, with different moments of inertia, without the need to manually change parameters. It should also show some robustness to when the mass center of the camera is not in the gimbal's rotational center.

- The control system should be able to function under the constraints imposed by the limited processing power of the control unit and the angular limitations of the gimbal.

1.2.2 Limitations

Aiming of the camera is a complex interaction between the gimbal and the platform. Turning the gimbal will, due to its size, create large disturbances, consisting of inertial forces on the platform. This will lead to a motion of the platform that will disturb the aiming of the gimbal that has to be compensated by the platform control system. However, the platform control system is considered to be a subject outside of this thesis so the platform will be assumed to be able to handle these disturbances. A simple test on the actual platform should however be made to get some knowledge of which motions that can be used without creating too much disturbance on the platform. Control strategies where the platform itself is used to gain better performance, e.g. faster pan motion over ground, will not be considered due to this simplification. A feed-forward signal from the gimbal to the platform may be discussed to be used in later development of the platform control system.

Another simplification is that the gimbal is viewed as a system of rigid bodies.

1.2.3 Changes from original goals

When the initial research and modelling was done (chapter 2) it was time to implement a control structure in a simulation environment (chapter 3), together with the mathematical model of the gimbal (chapter 2). It was also time to create a similar control structure in the real gimbal's control unit. However, it took some time before all parts of the gimbal were delivered, including the actuators. This delayed the code development as well as the development of the simulation environment due to missing friction data. When the actuators finally arrived it became clear that they were much too weak and could not produce the power needed for a good control performance of the gimbal. It was however decided that the project would continue but without the weight from a camera. The next big delay in the project came when the Pulse-Width Modulation (PWM) ports of the control unit failed. As these were not functional again until several weeks later, this resulted in a loss of time and momentum in the project.

The thesis was initially meant to focus on the control of the gimbal and the model was not intended for use outside of simulation of control in joint space. Due to the problems mentioned above, the entire control structure had to be developed in the simulation environment, rather than directly on the gimbal control unit. Thus, the goal of the control structure became to get a good active gyro stabilization around $\alpha = (0, 0, 0)$, whereas the steering was much less prioritized and the targeting goal was abandoned. Also, there was no longer any focus on more advanced control structures such as LQR control or linearization.

1.3 Earlier work

There is not much written about modelling and control of three-axis gimbals. Most development of gimbal control is for military applications and is guarded by secrecy. It is also hard to find any useful information from the commercial uses of gimbals such as mobile camera platforms at big sporting events. Most public papers in this field are about two-axis gimbals.

1.3.1 Mechanical modelling

In literature there are mainly two approaches to gimbal model:
In Kwon et al. [2007] a practical approach is introduced where the gimbal is presented as series of unknown moments of inertia, dampers and stiffnesses. These parameters are identified experimentally.
In Skoglar [2002] and Parveen [2009] a more theoretical approach is used where the gimbal is viewed as a robotic system. The model is developed by a combination of forward and inverse kinematics and dynamic equations, using the Lagrange approach. Remaining unknown parameters are then determined experimentally.

1.3.2 Gimbal control

The most common control designs are based on PID (proportional, integral, derivative) controllers. An interesting implementation is presented in Kwon et al. [2007] where a PI^2 controller, combined with a lead compensator, is used: $G_s(s) = K_p \frac{(s+\omega_1)(s+\omega_2)}{s^2} a \frac{(s+\omega_3)}{(s+\omega_4)}$ where the extra I-parameter enables a high open-loop gain in the low frequency region.

Good stabilization results are achieved in Seong et al. [2006] where a LQG/LTR (Linear Quadratic Gaussian control with Loop Transfer Recovery) controller is used. The results are compared to results obtained using a Lead-PI controller which shows that their LQG/LTR controller is better, especially in terms of decoupling. The same writers have in Seong et al. [2007] modified the controller to handle windup, due to input saturation, with good results.
In this work, it may be necessary to handle unknown or ill-defined disturbances and parameters, as mentioned in first paragraph of section 1.2.1. In Smith et al. [1999] and Shtessel [1999] a sliding mode control (SMC) technique is used where the differential equations that describe the system are forced into special manifolds (sliding surfaces) and handled by high speed switching control functions. It can be noted that Shtessel [1999] is an article on a three-axis platform.

2
Modelling

A mathematical model of the gimbal is important to deriving control strategies. It is for instance very difficult to tune PID parameters directly on a real system without some knowledge about it. A common approach is to use step responses and base the parameters on these. However, tests on a physical system are often complicated and time consuming, especially when the system is nonlinear which demands tests in several linearization points. A simulation model environment makes the control design process easier and faster. There are also some model based control strategies where the model itself is a part of the control loop.

A model of the system is also important for correct interpretation of the sensor inputs.

The modelling of the gimbal is divided into *kinematics* and *dynamics*. The kinematics will be used to interpret the sensors and will also be a part of the dynamics which in turn will be used in a simulation model.

2.1 Gimbal Properties

This section is a presentation of the gimbal properties without going into much details.

2.1.1 Mechanics

The gimbal was mainly made of aluminium profiles and consisted of four bodies (body (0)-(3) in Figure 2.1 and 2.2) connected by three revolute joints. Each joint was driven by a dc-motor (not shown in the figures below) via a gearbox. The mounting point of the camera on body (3) in the θ_2 direction. The box on body (1) containded the electronics including the control card.

Body (0) was supposed to be attached to the platform via a dampening joint which was supposed to cancel out some of the disturbances, e.g. from wind and motor vibrations. However, the gimbal was not attached to a platform during this thesis so body (0) became the part of the gimbal which was held by hand or fixed during tests.

The dimensions (l_1, h_1, b_2 and h_3) in Figure 2.1 and 2.2, and the mass, mass moment of inertia and center of mass for each body were given by the CAD models of the gimbal.

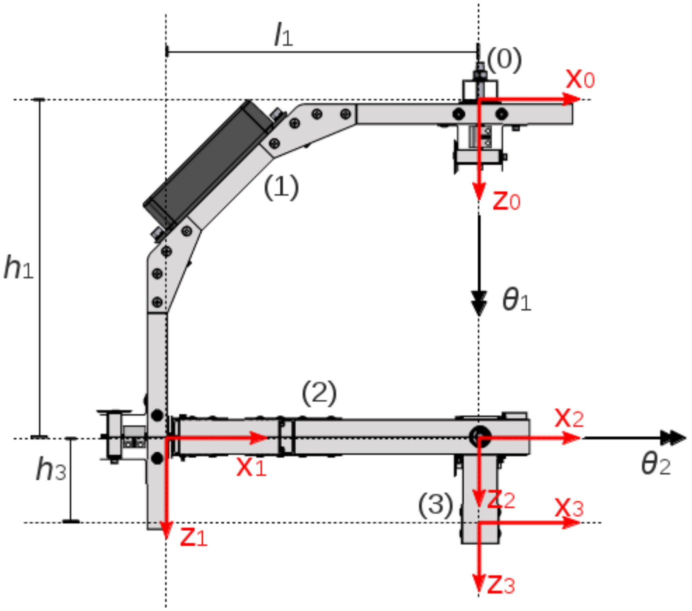


Figure 2.1: The gimbal in configuration $\theta = (0, 0, 0)$ viewed from the side with coordinate axes and distances between them. The direction of the joints is indicated by the double arrows.

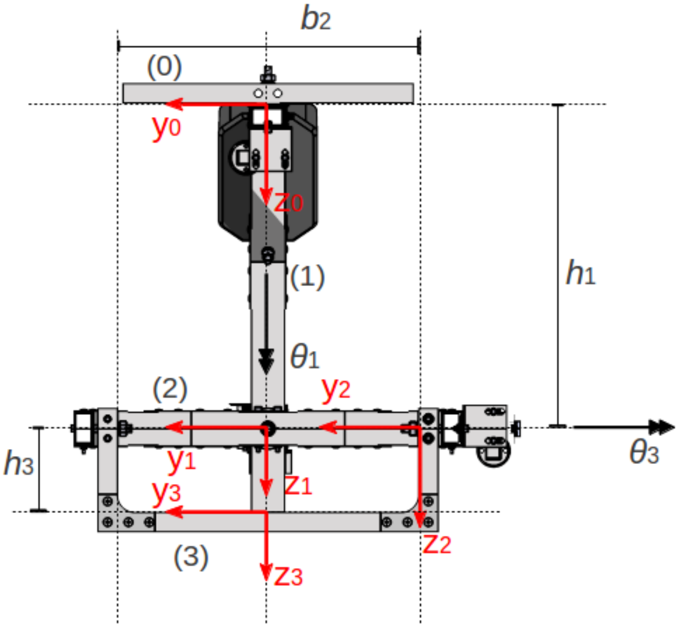


Figure 2.2: The gimbal in configuration $\theta = (0, 0, 0)$ *viewed from the front with coordinate axes and distances between them. The direction of the joints is indicated by the double arrows.*

2.1.2 Sensors

The gimbal was equipped with different sensors to aid the control:

Encoders

Rotary encoders were used to measure the angular position of the gimbal joints from the actuators. These were placed in the transmission between the dc-motors and the gimbal. The joint velocities were approximated using two angular positions and the time between the samples.

IMU

The gimbal was equipped with an Inertial Measurement Unit (IMU) close to the camera mounting point (origin of coordinate frame (3) in Figure 2.1). An IMU consists of gyroscopes (gyros) and accelerometers measuring accelerations and changes in attitude such as roll, pitch and yaw. The unit can with this information calculate the angular velocity of the device it is mounted on. The gyros (gyroscopes) were used to measure the angular velocities of the camera and these velocities were integrated to calculate the camera's angular position relative to the ground (attitude). However, a simple integration of the calculated velocities ("dead reckoning") also leads to accumulation of the errors present in them. The accumulated errors creates an ever-increasing difference between the calculated attitude and the actual attitude (drift). The drift is usually corrected using other sensors such as a GPS, which not available. A filter solution from IA using accelerometer data was used to get more correct attitudes (section 3.2.2).

2.1.3 User Input

User input to the gimbal consisted of two modes with either the desired angular velocities, or the desired attitude.

The first of these control modes was supposed to be used when the photographer uses a hand controller to control the yaw, pitch and roll of the camera in reference to its image output, i.e. the velocities are first defined in the camera's coordinate frame. The input from the hand controller consists three values that are interpreted as desired angular velocities of the camera. The gimbal control unit then calculates new reference velocities of the gimbal joints. When input ceases, so does the change in attitude.

The second input mode was a desired angular position relative to the ground (attitude). These angles are transformed to the gimbal's reference frames using the information from the IMU and new reference joint angles are given to the control.

2.2 Kinematics

The *kinematics* of a mechanical system is, according to Jazar [2007], the description of the system's translational and angular position and their time derivatives taking only geometry into account. The forces that cause the motion of the system are not taken into consideration but the kinematics can be used in the derivation of the dynamics of the system (section 2.3) which take into account all forces acting upon it. The basics of the kinematics of a rigid multi body system (MBS) is to describe the motion of a body in a coordinate frame attached in another body (e.g. a global fixed reference frame) using matrix algebra. The approach to do this is to combine a series of individual transformations between the body-attached coordinate systems. These transformations are a description of the relative rotation and distance between the bodies and consist of variable parameters according to the degrees of freedom (DOF) between the bodies. The gimbal in this thesis (see Figures 2.1 and 2.2) consists only of three revolute joints (R-joints) that individually have only one DOF (see Figure 2.3). This gives the whole MBS a total of three DOF, yaw, roll and pitch $(\theta_1, \theta_2, \theta_3)$. These are the *joint parameters* that henceforth will be used to describe the system. The kinematics can be divided into *forward* and *inverse kinematics*.

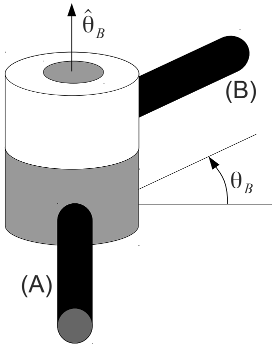


Figure 2.3: *Principal sketch of a revolute joint (R-joint) with index $[B]$ where body (B) is rotated an angle indicated by the joint parameter θ_B. It has in this case its zero value on the axis orthogonal to the length of body (A). $\hat{\theta}_B$ indicates the direction of the joint (the thumb in the right-hand-rule). A R-joint has only one degree of freedom and therefore only one joint parameter.*

2.2.1 Forward Kinematics

Forward kinematics of an MBS is how the *configuration* of the joint parameters affects the translational and angular position of an *end effector*. In robotics, the end effector is often the point where the robot's tools are attached and the "tool" of the gimbal was a camera. The forward kinematics can be described by transformation matrices. Serial robots are often described using the *Denavit-Hartenberg method* Jazar [2007][p. 199] which uses a sets of rules to systematically derive the forward kinematics in a standardized form. This method places the coordinate systems of each body according to the directions of the joints, for instance the z-axis parallel to the joint direction $\hat{\theta}$. However, since these coordinate systems and joint parameters seemed counterintuitive for this application, the method was abandoned in favour of for a more traditional rigid body mechanics method where the coordinate systems in the bodies are placed more freely.

Relation between axes and joint parameters

In Figures 2.1 and 2.2 there are two types of axes: the *coordinate axes* (x_i, y_i, z_i), indicated by an one-headed arrow, and the *joint parameter axes* $(\theta_1, \theta_2, \theta_3)$, indicated by the double-headed arrows. The joint parameters describe the angular position of the three joints in the gimbal, according to Figure 2.3. For the whole gimbal, motions in these joints are called yaw (θ_1), roll (θ_2) and pitch (θ_3). According to a common flight convention (Jazar [2007][p. 41]) yaw, roll, and pitch are defined such that yaw is a motion around a body's z-axis, roll around its x-axis and pitch around its y-axis. Each body can only move in one DOF so the joint parameters are defined in the following way: θ_1 as body (1)'s motion around the z_0-axis (yaw), θ_2 as body (2)'s motion around the x_1-axis (roll) and θ_3 as body (3)'s motion around the y_2-axis (pitch). The coordinate systems in the bodies of the gimbal (Figures 2.1 and 2.2) are placed in a way that they are parallel to each other in the configuration $(\theta_1, \theta_2, \theta_3) = (0, 0, 0)$. The forward kinematic matrices derived in this section are used in the dynamic modelling in section 2.3 as well as in the inverse kinematics described in section 2.2.2.

Homogeneous Transformations

The coordinate of an arbitrary point P of a rigid body in a local coordinate frame B is in this thesis denoted as the coordinate vector ${}^{B}\mathbf{r}_P$. The point can be described in a different coordinate frame G using the rotation matrix ${}^{G}\mathbf{R}_B$ and the translation vector ${}^{G}\mathbf{d}_B$ between the frames.

$$ {}^{G}\mathbf{r}_P = {}^{G}\mathbf{R}_B \, {}^{B}\mathbf{r}_P +^{G} \mathbf{d}_B, \tag{2.1}$$

where in a 3D environment

$$ {}^{B}\mathbf{r}_P = \begin{bmatrix} {}^{B}x_P \\ {}^{B}y_P \\ {}^{B}z_P \end{bmatrix}. \tag{2.2}$$

The expression (2.1) is called the *transformation* of the coordinate P from frame B to frame G.

It is often required to do several intermediate transformations to get the final transformation between two frames:

2.1 Example

The transformation between frame B_2 and frame B_0 is described by

$$^{0}\mathbf{r} = {}^{0}\mathbf{R}_2\, {}^{2}\mathbf{r} + {}^{0}\mathbf{d}_2, \tag{2.3}$$

where the components can be described using an intermediate coordinate frame B_1

$$^{0}\mathbf{R}_2 = {}^{0}\mathbf{R}_1\, {}^{1}\mathbf{R}_2 \tag{2.4}$$

$$^{0}\mathbf{d}_2 = {}^{0}\mathbf{R}_1\, {}^{1}\mathbf{d}_2 + {}^{0}\mathbf{d}_1. \tag{2.5}$$

The expressions for the transformations get very messy even if with relatively few intermediate frames. A convenient way to describe a transformation is to use *homogeneous transformation matrices* $^{G}\mathbf{T}_B$, where for a 3D environment

$$^{G}\mathbf{T}_B = \begin{bmatrix} {}^{G}\mathbf{R}_B & {}^{G}\mathbf{d}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{2.6}$$

The coordinate vector $^{B}\mathbf{r}$ that is being transformed must also be in a homogeneous form

$$^{B}_{h}\mathbf{r} = \begin{bmatrix} {}^{B}\mathbf{r} \\ 1 \end{bmatrix}. \tag{2.7}$$

2.2 Example

The transformation between frame B_2 and frame B_0 is described by

$$^{0}_{h}\mathbf{r} = {}^{0}\mathbf{T}_2\, {}^{2}_{h}\mathbf{r}, \tag{2.8}$$

where

$$^{0}_{h}\mathbf{r} = \begin{bmatrix} {}^{0}\mathbf{r} \\ 1 \end{bmatrix}, \quad {}^{2}_{h}\mathbf{r} = \begin{bmatrix} {}^{2}\mathbf{r} \\ 1 \end{bmatrix} \tag{2.9}$$

and

$$^{0}\mathbf{T}_2 = {}^{0}\mathbf{T}_1\, {}^{1}\mathbf{T}_2. \tag{2.10}$$

Gimbal transformation matrices

With the design parameters l_1, h_1, h_3 and b_2 in Figures 2.1 and 2.2 the transformations between the bodies are as follows.

The rotation matrix between the base frame (0) and the frame of body (1) is

$$^0\mathbf{R_1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

and distance between the coordinate origins is

$$^0\mathbf{d_1} = \begin{bmatrix} -l_1\cos(\theta_1) \\ -l_1\sin(\theta_1) \\ h_1 \end{bmatrix}. \tag{2.12}$$

The matrices (2.11) and (2.12) together create the homogeneous transformation matrix

$$^0\mathbf{T_1} = \begin{bmatrix} ^0\mathbf{R_1} & ^0\mathbf{d_1} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.13}$$

or

$$^0\mathbf{T_1} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & -l_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & -l_1\sin(\theta_1) \\ 0 & 0 & 1 & h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.14}$$

The transformation between the frames of body (1) and body (2):

$$^1\mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) \\ 0 & \sin(\theta_2) & \cos(\theta_2) \end{bmatrix} \tag{2.15}$$

$$^1\mathbf{d}_2 = \begin{bmatrix} l_2 \\ -\frac{1}{2}b_2\cos(\theta_2) \\ -\frac{1}{2}b_2\sin(\theta_2) \end{bmatrix} \tag{2.16}$$

$$^1\mathbf{T}_2 = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) & -\frac{1}{2}b_2\cos(\theta_2) \\ 0 & \sin(\theta_2) & \cos(\theta_2) & -\frac{1}{2}b_2\sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.17}$$

The transformation between the frames of body (2) and body (3):

$$^2\mathbf{R}_3 = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) \\ 0 & 1 & 0 \\ -\sin(\theta_3) & 0 & \cos(\theta_3) \end{bmatrix} \tag{2.18}$$

$$
{}^{2}\mathbf{d}_3 = \begin{bmatrix} h_3\sin(\theta_3) \\ \frac{1}{2}b_2 \\ h_3\cos(\theta_3) \end{bmatrix} \tag{2.19}
$$

$$
{}^{2}\mathbf{T}_3 = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & h_3\sin(\theta_3) \\ 0 & 1 & 0 & \frac{1}{2}b_2 \\ -\sin(\theta_3) & 0 & \cos(\theta_3) & h_3\cos(\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.20}
$$

The total transformation between body (3) and the base is

$$
{}^{0}\mathbf{T}_3 = \begin{bmatrix} {}^{0}\mathbf{R}_3 & {}^{0}\mathbf{d}_3 \\ \mathbf{0} & 1 \end{bmatrix} = {}^{0}\mathbf{T}_1{}^{1}\mathbf{T}_2{}^{2}\mathbf{T}_3, \tag{2.21}
$$

where, with the notation $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$,

$$
{}^{0}\mathbf{R}_3 = \begin{bmatrix} c_1c_3 - s_1s_2s_3 & -c_2s_1 & c_1s_3 + c_3s_1s_2 \\ c_3s_1 + c_1s_2s_3 & c_1c_2 & s_1s_3 - c_1c_3s_2 \\ -c_2s_3 & s_2 & c_2c_3 \end{bmatrix} \tag{2.22}
$$

and

$$
{}^{0}\mathbf{d}_3 = \begin{bmatrix} c_1l_2 - c_1l_1 + c_1h_3s_3 + c_3h_3s_1s_2 \\ l_2s_1 - l_1s_1 + h_3s_1s_3 - c_1c_3h_3s_2 \\ h_1 + c_2c_3h_3 \end{bmatrix}. \tag{2.23}
$$

${}^{0}\mathbf{R}_3$ is a pitch-roll-yaw rotation matrix. It can also more generally be called a ZXY *triple rotation* matrix Jazar [2007][p. 40] due to the order in which the rotation matrices are successively multiplied. These transformations are used in the derivation of the *inverse kinematics* in section 2.2.2 as well as the dynamics in section 2.3.

One other transformation that is used in the Inverse kinematics in section 2.2.2 is the rotation between body (3) and the ground, viewed as an imaginary body called (g). The angular position of body (3) is derived using the information from the gyros and accelerometers on the IMU attached to the body which are represented by a pitch-roll-yaw matrix, similar to matrix (2.22).

$$
{}^{g}\mathbf{R}_3 = \begin{bmatrix} c\alpha_1c\alpha_3 - s\alpha_1s\alpha_2s\alpha_3 & -c\alpha_2s\alpha_1 & c\alpha_1s\alpha_3 + c\alpha_3s\alpha_1s\alpha_2 \\ c\alpha_3s\alpha_1 + c\alpha_1s\alpha_2s\alpha_3 & c\alpha_1c\alpha_2 & s\alpha_1s\alpha_3 - c\alpha_1c\alpha_3s\alpha_2 \\ -c\alpha_2s\alpha_3 & s\alpha_2 & c\alpha_2c\alpha_3 \end{bmatrix}, \tag{2.24}
$$

where $s\alpha_i$ and $c\alpha_i$ are short for $\sin(\alpha_i)$ and $\cos(\alpha_i)$ and α_1 = yaw, α_2 = roll and α_3 = pitch derived using the information from the IMU. There exist a total of 12 possible independent orientation representation of this type, but this one was chosen due to its similarity to matrix (2.22).

2.2.2 Inverse Kinematics

The opposite of forward kinematics is when the joint parameters are determined from the translational and angular position of the end-effector. This is called *inverse kinematics*. In a common robotic application the goal is to move a tool to a certain position and hold it in a certain orientation. This is often a difficult problem because it leads to multiple solutions which are often avoided by letting the robot "learn" the configuration of the joint parameters instead of computing them. Inverse kinematics in the gimbal application is simpler, because in this case it is only concerned with the orientation (in this thesis called angular position) of the camera. Thus multiple solutions of trigonometric functions can be avoided. The inverse kinematics will be used to generate the reference signals to the gimbal control.

Angular position

According to the second user input mode described in section 2.1.3 one goal is to change the attitude of the camera relative to the ground. This will result in a control error between the desired attitude and with gyros and accelerometers derived current attitude described by (2.24). This can be presented as a new rotation matrix in the camera's reference frame.

$$ {}^{3}\mathbf{R}_e = \begin{bmatrix} c\varepsilon_1 c\varepsilon_3 - s\varepsilon_1 s\varepsilon_2 s\varepsilon_3 & -c\varepsilon_2 s\varepsilon_1 & c\varepsilon_1 s\varepsilon_3 + c\varepsilon_3 s\varepsilon_1 s\varepsilon_2 \\ c\varepsilon_3 s\varepsilon_1 + c\varepsilon_1 s\varepsilon_2 s\varepsilon_3 & c\varepsilon_1 c\varepsilon_2 & s\varepsilon_1 s\varepsilon_3 - c\varepsilon_1 c\varepsilon_3 s\varepsilon_2 \\ -c\varepsilon_2 s\varepsilon_3 & s\varepsilon_2 & c\varepsilon_2 c\varepsilon_3 \end{bmatrix}, \tag{2.25} $$

where $\varepsilon_1,\ \varepsilon_2$ and ε_3 are the errors ($\varepsilon = \alpha_{des} - \alpha$) of yaw, roll and pitch in the camera's reference frame. e denotes an imagined error coordinate frame.

The total rotation matrix between the error frame (e) and the gimbal base frame (0) is then

$$ {}^{0}\mathbf{R}_e(\Theta, \varepsilon) = {}^{0}\mathbf{R}_3(\Theta)\, {}^{3}\mathbf{R}_e(\varepsilon), \tag{2.26} $$

where Θ are the current angles of the joints ($\theta_1,\ \theta_2,\ \theta_3$) (Figures 2.1, 2.2 and 2.3).

A new joint configuration Θ_{new} is chosen such that

$$ {}^{0}\mathbf{R}_3(\Theta_{new}) = {}^{0}\mathbf{R}_e(\Theta, \varepsilon), \tag{2.27} $$

which means that coordinate frame (3) becomes in the same frame as (e), i.e the desired attitude is reached.

Equation (2.27) gives together with (2.26)

$$^0\mathbf{R}_3(\Theta_{new} =^0 \mathbf{R}_3(\Theta)\ ^3\mathbf{R}_e. \tag{2.28}$$

If the right hand side of (2.28) is expressed as

$$^0\mathbf{R}_3(\Theta)\ ^3\mathbf{R}_e = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{31} \end{bmatrix}, \tag{2.29}$$

where r_{ij} is the matrix component ij of $^0\mathbf{R}_3(\Theta)\ ^3\mathbf{R}_e$, then (2.28) together with (2.22), in section 2.2.1, gives the new joint parameters:

$$\theta_{1new} = \text{arctan2}(-r_{12}, r_{22}), \tag{2.30a}$$
$$\theta_{2new} = \text{arcsin}(r_{32}), \tag{2.30b}$$
$$\theta_{3new} = \text{arctan2}(-r_{31}, r_{33}), \tag{2.30c}$$

where arctan2(y,x) is a function that places the computed angle in the correct quadrant of the unit circle.

These angles will be the angular position reference signals for the individual joints.

Angular velocity

According to the first user input mode described in section 2.1.3, the input can be a set of desired angular velocities Φ of the camera in its own coordinate frame:

$$^3_3\Phi = \begin{bmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{bmatrix} = \begin{bmatrix} \varphi_{roll} \\ \varphi_{pitch} \\ \varphi_{yaw} \end{bmatrix}. \tag{2.31}$$

The relationship between the local angular velocities Ω in the joints and the camera angular velocity can be expressed using a *Jacobian matrix* **J**

$$^3_3\Phi = \mathbf{J}\Omega, \tag{2.32}$$

where

$$\Omega = (\omega_1,\ \omega_2,\ \omega_3)^T = (\dot\theta_1,\ \dot\theta_2,\ \dot\theta_3)^T. \tag{2.33}$$

As in Jazar [2007][p. 351] the Jacobian $\mathbf{J}$ can be viewed as the angular velocity directions of the joints presented in the camera's coordinate frame and because the gimbal consists only of R-joints it can be written as

$$\begin{aligned}\mathbf{J} &= \begin{bmatrix} {}^3\hat{\theta}_1 & {}^3\hat{\theta}_2 & {}^3\hat{\theta}_3 \end{bmatrix} \\ &= \begin{bmatrix} {}^3\mathbf{R}_1\,{}^1\hat{\theta}_1 & {}^3\mathbf{R}_2\,{}^2\hat{\theta}_2 & {}^3\hat{\theta}_3 \end{bmatrix} \\ &= \begin{bmatrix} \left({}^1\mathbf{R}_2\,{}^2\mathbf{R}_3\right)^T\,{}^1\hat{\theta}_1 & {}^2\mathbf{R}_3^T\,{}^2\hat{\theta}_2 & {}^3\hat{\theta}_3 \end{bmatrix},\end{aligned} \tag{2.34}$$

where

$${}^1\hat{\theta}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad {}^2\hat{\theta}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad {}^3\hat{\theta}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \tag{2.35}$$

which are the directions of the joints (Figures 2.1, 2.2 and 2.3).

(2.32) is then

$$\begin{bmatrix} \varphi_{roll} \\ \varphi_{pitch} \\ \varphi_{yaw} \end{bmatrix} = \begin{bmatrix} -\cos(\theta_2)\sin(\theta_3) & \cos(\theta_3) & 0 \\ \sin(\theta_2) & 0 & 1 \\ \cos(\theta_2)\cos(\theta_3) & \sin(\theta_3) & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \tag{2.36}$$

which is then reordered to

$$\begin{bmatrix} \varphi_{yaw} \\ \varphi_{roll} \\ \varphi_{pitch} \end{bmatrix} = \begin{bmatrix} \cos(\theta_2)\cos(\theta_3) & \sin(\theta_3) & 0 \\ -\cos(\theta_2)\sin(\theta_3) & \cos(\theta_3) & 0 \\ \sin(\theta_2) & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}. \tag{2.37}$$

The desired joint velocities Ω can then be expressed as

$$\Omega = \mathbf{J}^{-1} \begin{bmatrix} \varphi_{yaw} \\ \varphi_{pitch} \\ \varphi_{roll} \end{bmatrix}. \tag{2.38}$$

It is also possible to derive the current *translational* velocity $\dot{\mathbf{X}}$ of the end-effector with another Jacobian matrix $\mathbf{J_d}$

$$\dot{\mathbf{X}} = \mathbf{J_D}\Omega = \begin{bmatrix} {}^3\hat{\theta}_0 \times {}^\mathbf{3}\mathbf{d_1} & {}^3\hat{\theta}_1 \times {}^\mathbf{3}\mathbf{d}_2 & {}^3\hat{\theta}_2 \times {}^\mathbf{3}\mathbf{d_3} \end{bmatrix}\Omega. \tag{2.39}$$

This can be useful if the translational velocity is needed, e.g. in applications such as positional tracking and sensor filtering. Translational Jacobians are used in the *dynamics* in section 2.3.

2.3 Dynamics

The *dynamics* of an MBS is how its motion is affected by external forces such as gravity and torque from the joint actuators. It is presented as a system of differential equations obtained using either *Newton-Euler equations* Jazar [2007][p. 447], which use Newton's second law of motion, or *Lagrange equations*
Jazar [2007][p. 478], which uses the kinetic and potential energies of the MBS. The method using Lagrange equations was chosen in this thesis. The reason for this is mainly the writer's desire to learn this method.

2.3.1 Lagrange Dynamics

A systematic way to obtain the equations of motion of a robot with n joints is according to Jazar [2007][p. 528] to use the Lagrange equation:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \omega_i}\right) - \frac{\partial \mathcal{L}}{\partial \theta_i} = Q_i \quad i = 1, 2, \ldots n. \tag{2.40}$$

$\mathcal{L}$ is the *Lagrangian* which is defined as the difference between the kinetic energy K and potential energy V

$$\mathcal{L} = K - V. \tag{2.41}$$

θ_i are the coordinates of the system (which in the gimbal application are the current angles in joints $i = 1, 2, 3$). ω_i is the time derivative $\dot{\theta}_i$ of the coordinates and Q_i is the non-potential force driving θ_i. Q_i are external forces such as torques from actuators while internal forces such as gravity and Coriolis effects Jazar [2007][p. 451] are already a part of the left hand side of equation (2.40).

According to the proof in Jazar [2007][p. 529] the kinetic energy of the whole robot can be expressed as

$$K = \frac{1}{2}\mathbf{\Omega}_j^T D \mathbf{\Omega}_j \quad j = 1, 2, \ldots n, \tag{2.42}$$

where $\Omega = (\omega_1, \omega_1, \omega_3)^T$ and D is the $n \times n$ matrix

$$D = \sum_{j=1}^{n} \left(\mathbf{J}_{Dj}^T \, m_j \, \mathbf{J}_{Dj} + \frac{1}{2} \mathbf{J}_{Rj}^T \, {}^0 I_j \, \mathbf{J}_{Rj} \right), \tag{2.43}$$

which is called the inertial-type matrix. $\mathbf{J}_{Dj}$ and $\mathbf{J}_{Rj}$ are the *Jacobians* of body (j) that transform the joint angular velocities into the translational and rotational velocities of the body's center of mass (CM) (similar to the jacobians in section 2.2.2). The component m_j is the mass of body (j) and ${}^0I_j$ is the *Mass moment of inertia matrix* about its CM and expressed in the base coordinate frame (0).

The potential energy can according to the same proof [Jazar, 2007, p. 529] be written as

$$V = \sum_{i=1}^{n} V_i = -\sum_{i=1}^{n} m_i \, {}^0\mathbf{g}^T \, {}^0\mathbf{r}_i, \tag{2.44}$$

where ${}^0\mathbf{g}$ is the gravitational acceleration vector in the base frame (0) and ${}^0\mathbf{r}_i$ are the coordinates of the center of mass of each body expressed in the base frame (0).

If equations (2.41) to (2.44) are inserted in (2.40) the result can with some rearrangement (done in Jazar [2007][p. 529]) and $\Theta = (\theta_1, \theta_2, \theta_3)^T$ be presented in a matrix form

$$\mathbf{D}(\Theta)\dot{\Omega} + \mathbf{H}(\Theta, \Omega) + \mathbf{G}(\Theta) = \mathbf{Q}, \tag{2.45}$$

or in summation form

$$\sum_{j=1}^{n} D_{ij}(\theta)\dot{\omega}_j + \sum_{k=1}^{n}\sum_{l=1}^{n} H_{ikl}\omega_k\omega_l + G_i = Q_i. \tag{2.46}$$

The index $i = 1, 2, \dots n$ indicates the row of the equation.

Vector $\mathbf{H}$ is called the *Velocity coupling vector* and is

$$\begin{aligned} H_i &= \sum_{k=1}^{n}\sum_{l=1}^{n} H_{ijk}\omega_k\omega_l \\ &= H_{i11}\omega_1\omega_1 + \ldots + H_{in1}\omega_n\omega_1 + \ldots + H_{inn}\omega_n\omega_n, \end{aligned} \tag{2.47}$$

$$\tag{2.48}$$

where

$$H_{ijk} = \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial D_{ij}}{\partial \theta_k} - \frac{1}{2}\frac{\partial D_{jk}}{\partial \theta_i}\right). \tag{2.49}$$

The last term in (2.46) is the *Gravitational vector* G_i

$$G_i = \sum_{j=1}^{n} m_j \, {}^0\mathbf{g}^T \mathbf{J}_{Dj}^{(i)}. \tag{2.50}$$

Equation (2.46) can in theory be viewed as how reaction torques $\mathbf{Q}$ in each joint are affected. The $n \times n$ *inertial-type matrix* $\mathbf{D}$ is how the reaction torques are affected by a given set of accelerations $\dot{\mathbf{\Omega}}$ due to the inertias of the bodies and every component outside the diagonal are coupling effects. The $n \times 1$ *Velocity coupling vector* $\mathbf{H}$ contains the reactions due to a given set of angular velocities $\mathbf{\Omega}$ and the bodies' inertias and coupling effects. $\mathbf{H}$ can often be referred as a *Christoffel operator* Jazar [2007][p. 533]. The $n \times 1$ *Gravitational vector* $\mathbf{G}$ contains the reaction torques in the joints due to the gravitational pull ${}^0\mathbf{g}$.

Lagrange using link transformation matrices

Instead of deriving each Jacobian that is needed in the terms of (2.46) the Lagrange equations can be derived using the link transformation matrices ${}^0\mathbf{T}_r$ derived in section 2.2.

It is proven in Jazar [2007][p. 535] that the terms in (2.46) can be rewritten such that the *Inertial-type matrix* (2.43) becomes

$$D_{ij} = \sum_{r=max(i,j)}^{n} tr\left(\frac{\partial^0\mathbf{T}_r}{\partial\theta_i} \, {}^r\bar{I}_r \left(\frac{\partial^0\mathbf{T}_r}{\partial\theta_j} \right)^T \right), \tag{2.51}$$

the *Velocity coupling* (2.49)

$$H_{ijk} = \sum_{r=max(i,j,k)}^{n} tr\left(\frac{\partial^2 \, {}^0\mathbf{T}_r}{\partial\theta_i\partial\theta_k} \, {}^r\bar{I}_r \left(\frac{\partial^0\mathbf{T}_r}{\partial\theta_i} \right)^T \right), \tag{2.52}$$

and the *Gravitational vector* (2.50)

$$G_i = -\sum_{r=i}^{n} m_r \, {}^0\mathbf{g}^T \frac{\partial^0\mathbf{T}_r}{\partial\theta_i} \, {}^r\mathbf{r}_r. \tag{2.53}$$

The $tr()$ indicates the trace function of the matrices and $\mathbf{r}_r$ is the translational position of the center of mass of body (r) expressed in frame (r).

Matrix ${}^r\bar{I}_r$ is the *pseudo inertia matrix* of body r expressed in r's reference frame which is compatible with the homogeneous transformation matrices, see Jazar [2007][p. 467].

$$\bar{I} = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & I_{xy} & I_{xz} & mx_{cm} \\ I_{yx} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & I_{yz} & my_{cm} \\ I_{zx} & I_{zy} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & mz_{cm} \\ mx_{cm} & my_{cm} & mz_{cm} & m \end{bmatrix} \tag{2.54}$$

where x_{cm} is the x-coordinate of the body's center of mass. I_{ii} are the elements of the mass moment of inertia $\mathbf{I}_r$ of body (r), Jazar [2007][p. 466].

2.3.2 Dynamics of the gimbal

One difference between the gimbal application and most robotic applications is the gravitational vector ${}^0\mathbf{g}$. The direction of it is in most applications constant straight down (along the global z-axis). The gimbal is however supposed to be in flight which means that ${}^0\mathbf{g}$ varies with the attitude of platform. The gravitational vector $(0, 0, g)^T$ on the ground is therefore transformed to the base frame (0) via gyro and accelerometer data from the IMU.

$$ {}^0\mathbf{g} = {}^0\mathbf{R}_3 \, {}^g\mathbf{R}_3^T \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = {}^0\mathbf{R}_3 \begin{bmatrix} -c\alpha_2 s\alpha_3 \\ s\alpha_2 \\ c\alpha_2 c\alpha_3 \end{bmatrix} g \tag{2.55} $$

where ${}^0\mathbf{R}_3$ is (2.22), ${}^g\mathbf{R}_3^T$ the inverse of (2.24) and g is the gravitational acceleration ($9.81 m/s^2$). This gives that $\mathbf{G}$ in (2.45) becomes a function of both the joint parameters Θ and the IMU attitude α, $\mathbf{G}(\Theta, \alpha)$.

With the aid of Matlab symbolic toolbox and Mathematica, the terms $\mathbf{D}(\Theta)\dot{\Omega}$, $\mathbf{H}(\Theta, \Omega)$ and $\mathbf{G}(\Theta, \alpha)$ were obtained using the kinematics in section 2.2. The design parameters described in section 2.1, such as masses, lengths and moments of inertia I were taken from the CAD-models of the gimbal.

The Lagrange equations (2.45) were rearranged to

$$ \dot{\Omega} = \mathbf{D}^{-1}(\mathbf{Q} - \mathbf{H} - \mathbf{G}) \tag{2.56} $$

and a state vector was defined to describe the dynamics

$$ \mathbf{x} = \begin{bmatrix} \Omega \\ \Theta \end{bmatrix} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}. \tag{2.57} $$

With (2.56) and one derivation of (2.57) the system of differential equations that describes the dynamics of the gimbal were obtained

$$ \dot{\mathbf{x}} = \begin{bmatrix} \mathbf{D}(\mathbf{x}_{4:6})^{-1}(\mathbf{Q} - \mathbf{H}(\mathbf{x}) - \mathbf{G}(\mathbf{x}_{4:6}, \alpha)) \\ \mathbf{x}_{1:3} \end{bmatrix}. \tag{2.58} $$

The IMU attitude α could have been a set of states among (2.57) but is not directly a part of the joint states and is therefore viewed as a separate set of states.

Camera mechanics

A camera attached to the gimbal is in the dynamic calculations viewed as a part of body (3), Figure 2.1. If the camera is changed the mass m_3, center of mass coordinates ${}^3\mathbf{r}_3$ and the mass moment of inertia $\mathbf{I}_3$ of body (3) will also change. The new mass of the body is simply the addition of the masses of the camera and the original body:

$$m_3 = m_{3org} + m_c. \tag{2.59}$$

According to Jazar [2007][p. 449] the combined center of mass of a rigid body B is defined as

$${}^B\mathbf{r}_{cm} = \frac{1}{m}\int_B \mathbf{r}\,\mathrm{d}m. \tag{2.60}$$

If the masses m_i and the center of masses $\mathbf{r}_i = (x_{cm}, y_{cm}, z_{cm})^T$ are known for each part of the body, equation (2.60) can be written as

$${}^B\mathbf{r}_{cm} = \frac{1}{\sum m_i}\sum \mathbf{r}_i m_i, \tag{2.61}$$

where the summations are of each body (i) incuded in body B.

In the gimbal application the center of mass of body (3) with a camera attached becomes

$${}^3\mathbf{r}_3 = \frac{1}{m_3}(\mathbf{r}_{3org} m_{3org} + \mathbf{r}_c m_c). \tag{2.62}$$

Vector $\mathbf{r}_{3org}$ is the coordinate of the original body's center of mass expressed in the coordinate frame (3) (see Figure 2.2) and $\mathbf{r}_c$ is the center of mass of the camera expressed in coordinate frame (3).

The total mass moment of inertia of body (3) is obtained using the *parallel-axes theorem,* also called *Huygens-Steiner's theorem* (Jazar [2007][p. 473]).

$$\mathbf{I}_3 = \mathbf{I}_{3org} + m_{3org}\tilde{\mathbf{r}}_{3org}\tilde{\mathbf{r}}_{3org}^T + \mathbf{I}_c + m_c\tilde{\mathbf{r}}_c\tilde{\mathbf{r}}_c^T, \tag{2.63}$$

where

$$\tilde{\mathbf{r}}_i = \begin{bmatrix} 0 & -z_{cm} & y_{cm} \\ z_{cm} & 0 & -x_{cm} \\ -y_{cm} & x_{cm} & 0 \end{bmatrix}. \tag{2.64}$$

$\tilde{\mathbf{r}}_i$ is the *skew-symmetric matrix* form of $\mathbf{r}_i$.

The shape of the camera can be simplified to be a cuboid and the mass moment of inertia can be looked up in a table of moments of inertias. If a camera objective is used the total camera properties can be calculated as in equations (2.59), (2.62) and (2.63).

The same theory can be applied if other parts of the gimbal are not included in the original body. It can for instance be useful to treat the motors of the gimbal as separate bodies if simulation tests of different actuators (motors and gearboxes) are desired. Instead of generating a whole new model, with a different set of actuators or a different camera, their mechanical properties are treated as symbolic quantities, and given appropriate values in the simulation environment. The model in this thesis allows for interchangeable cameras, but not actuators, as these are included in the original gimbal bodies.

Friction

The torques $\mathbf{Q}$ on the gimbal are affected by friction which is assumed to be linear

$$\mathbf{Q} = \tau_{\mathbf{a}} - \tau_f sgn(\Omega) - \mu\Omega \tag{2.65}$$

where $\tau_{\mathbf{a}}$ are the torques from the actuators, τ_f the static frictions in the joints and μ is a diagonal matrix containing the kinetic friction coefficients of the joints.

The combined torques in a joint need in reality overcome the static friction τ_f in order to create a motion. The condition for the gimbal to behave according to (2.58) from a standstill is therefore according to Algorithm 2.1.

Algorithm 2.1 Condition for motion in joint i due to static friction

```
if $\left(\left|\left[\tau_{\mathbf{a}} - \mathbf{D}\dot{\Omega} - \mathbf{H} - \mathbf{G})\right]_i\right| < \tau_f(i)\right)$ && $(\omega_i == 0)$ then
    $\dot{\omega}_i = 0;$
else
    $\dot{\omega}_i = \left[\mathbf{D}^{-1}(\mathbf{Q} - \mathbf{H} - \mathbf{G})\right]_i;$
end if
```

Actuator dynamics

According to Mohan et al. [2003] dc-motor drives can be described using the following equations:

$$\tau_m = K_t i, \tag{2.66}$$

where τ_m is the output torque from the motors, K_t the torque constant and i the current. The current is expressed in the differential equation

$$\frac{di}{dt} = \frac{1}{L}(u - R_m i - e), \tag{2.67}$$

where u is the input voltage, R_m the resistance of the motor circuit, e the back-emf of the motor and L the motors inductance. e is produced by the angular velocity ω_m

$$e = K_e \omega_m, \tag{2.68}$$

where K_e the back-emf constant.

With the coupling ratio $a = \frac{\omega_m}{\omega} = \frac{\tau_a}{\tau_m}$ of the gearboxes the equations of the actuators becomes

$$\tau_a = K_t a i \tag{2.69}$$

and

$$\frac{di}{dt} = \frac{1}{L}(u - R_m i - K_e a \omega) \tag{2.70}$$

The constants K_t, K_e L, R_m and a are taken from the data sheets of the motor and the gearbox.

2.4 Identification

If the data from the CAD-models, mentioned in section 2.3.2, is assumed to be correct, the only unknown parameters in equations (2.58) to (2.70) that need to be estimated are the friction parameters τ_f and μ for each axis. The noise in the gimbal's sensors was identified in this section and was later added as disturbance in the simulation environment.

2.4.1 Friction tuning

In order to estimate the friction in each axis a series of voltage step response tests were made. The results from these tests were compared to the results from the same steps in the simulation environment. The static and dynamic friction parameters in the model, τ_f and μ mentioned in section 2.3.2, were then simply tuned to get a simulation result that fitted the results from the real measurements. All tests were done around $\theta = (0, 0, 0)$.

Figure 2.4 shows that the simulation results were quite close to the real measurements but the overall angular velocity was too high. The static friction parameter τ_f was tuned to reduce the overall angular velocity and the dynamic friction parameter μ was tuned to set a penalty on higher angular velocities. The result is shown in Figure 2.5.

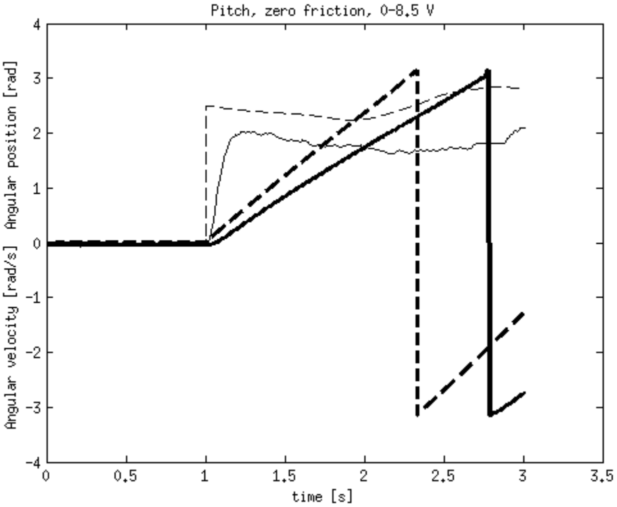


Figure 2.4: *Step response on the pitch axis with friction parameters set to zero. Voltage step from 0 to 8.5 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

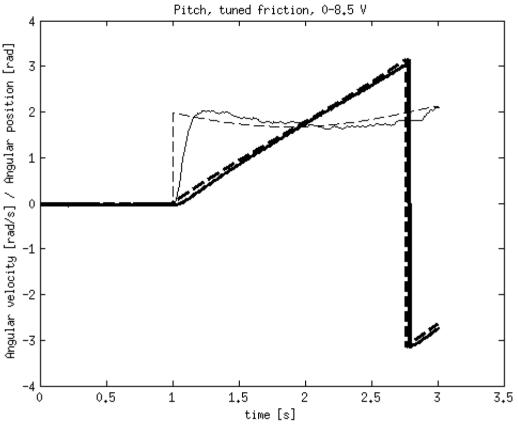


Figure 2.5: *Step response on the pitch axis with tuned friction parameters. Voltage step from 0 to 8.5 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

The behaviour was somewhat different when the voltage to the motors were reversed, as shown in Figure 2.6. Negative directions of the torque on the axes needed separate sets of friction parameters. The results of friction tuning in the negative direction on the pitch axis is shown in Figure 2.7.

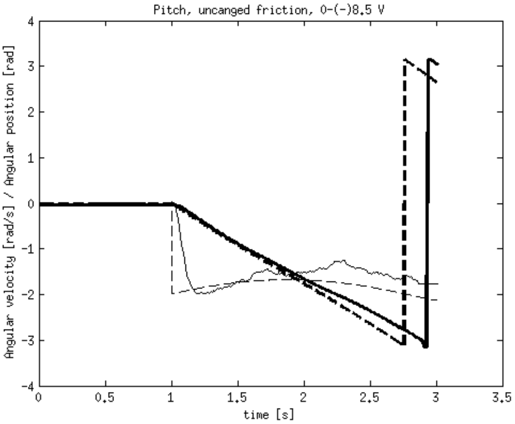


Figure 2.6: *Negative step response on the pitch axis with same friction parameters as the positive step. Voltage step from 0 to -8.5 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

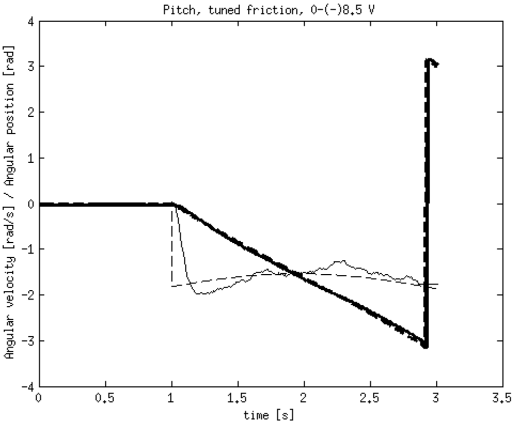


Figure 2.7: *Negative step response on the pitch axis with tuned friction parameters. Voltage step from 0 to -8.5 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

The results from the roll and yaw axes are shown in Figure 2.8 to 2.13.

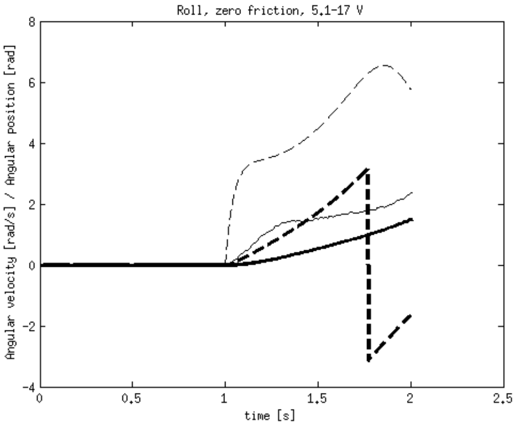


Figure 2.8: *Step response on the roll axis with friction parameters set to zero. Voltage step from 5.1 to 17 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

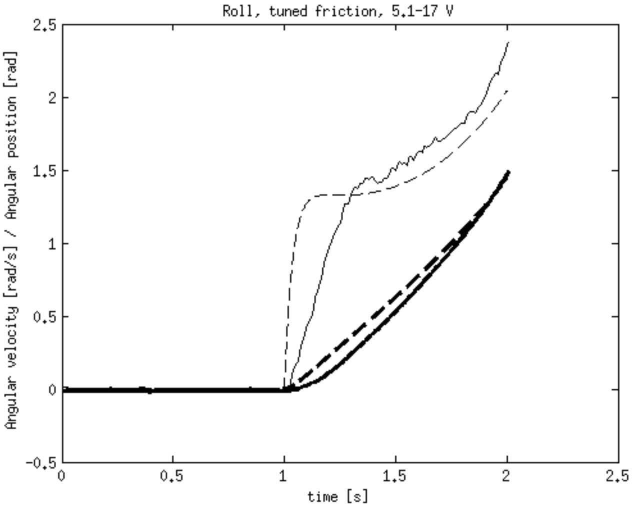


Figure 2.9: *Step response on the roll axis with with tuned friction parameters. Voltage step from 5.1 to 17 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

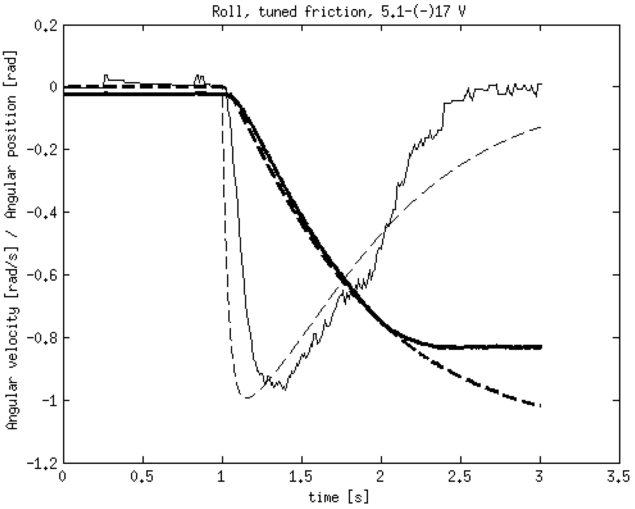


__Figure 2.10:__ Negative step response on the roll axis with with tuned friction parameters. Voltage step from 5.1 to -17 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.

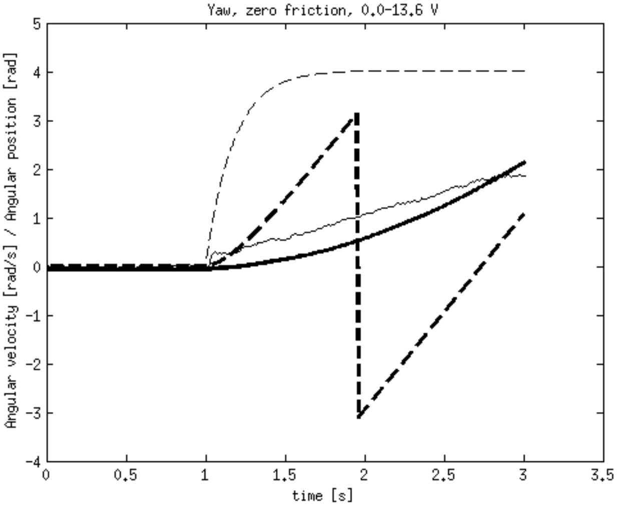


__Figure 2.11:__ Step response on the yaw axis with friction parameters set to zero. Voltage step from 0 to 13.6 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.

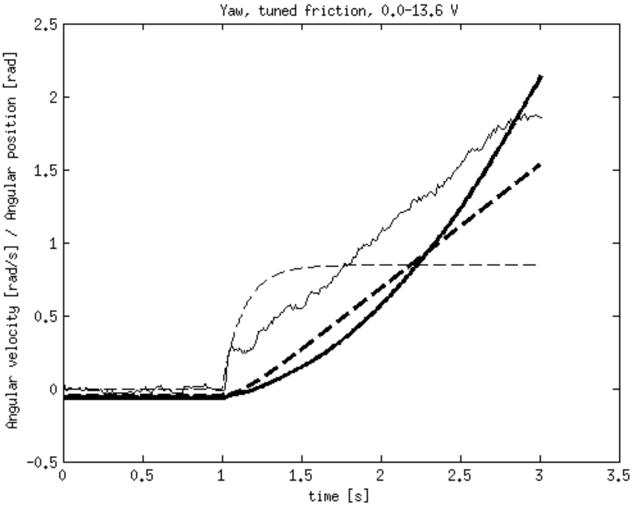


Figure 2.12: *Step response on the yaw axis with tuned friction parameters. Voltage step from 0 to 13.6 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

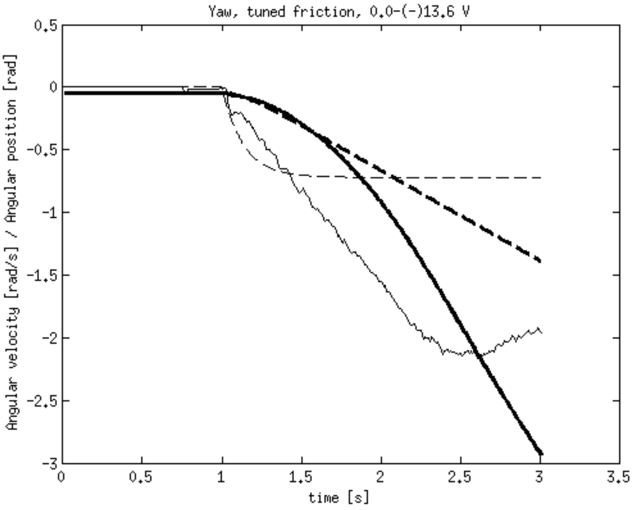


Figure 2.13: *Negative step response on the yaw axis with tuned friction parameters. Voltage step from 0 to -13.6 V. Thin lines are angular velocity in rad/s and thick lines angular position in radians. Solid lines are real results and dashed lines are simulation results.*

The results on the yaw axis and to some extent the other axes shows that the assumption in (2.65) was not sufficient to represent the friction in the joints. The bearings in the yaw joint were too small to distribute the load from the gimbal evenly witch results in a nonlinear behaviour and dependency on the joint angular position. The friction parameters in yaw were tuned to at least get an accurate result around $\theta_1 = 0$. Tests with a friction model where there was only dynamic friction when the joint was in motion did not yield any better results. The friction parameters are presented in Table 2.1.

Table 2.1: *Friction in the gimbal joints. (+) indicates forward direction and (-) negative direction.*

Axis	$\tau_{\mathbf{f}}[Nm]$	$\mu[Nm/(rad/s)]$
Yaw (+)	0.45	4.5e-2
Yaw (-)	0.47	4.5e-2
Roll (+)	0.28	3.5e-2
Roll (-)	0.12	2.5e-2
Pitch (+)	0.05	1.5e-2
Pitch (-)	0.05	3.0e-2

Alternative method

Alternatively, instead of simple tuning of the friction parameters, would be to use measurements on the axis and together with the dynamic equations above identify the friction.

A series of the angular accelerations $\dot{\Omega}$ can first be approximated from the measured series of the angular velocities Ω mentioned in section 2.4.1. The angular velocities Ω and the known voltage u can then be inserted into equations (2.69) and (2.70) to calculate the current actuator torque τ_a at each sampled time instance. The measured variables Ω and Θ and the calculated variables $\dot{\Omega}$ and τ_a be inserted into equations (2.45) and (2.65) to get the relationship

$$\mathbf{D}(\Theta)\dot{\Omega} + \mathbf{H}(\Theta, \Omega) + \mathbf{G}(\Theta) - \tau_a = \mathbf{f}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}}), \tag{2.71}$$

where the function $\mathbf{f}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}})$ is how the friction (and potential model errors) depend on the angular position, velocity and acceleration.

The left hand side of (2.71) and the known variables can then be inserted into some identification software, such as the System Identification toolbox in Matlab, to identify $\mathbf{f}(\boldsymbol{\Theta}, \boldsymbol{\Omega}, \dot{\boldsymbol{\Omega}})$. Measurements where the input voltage u is a pulse train that changes with different amplitudes and frequencies would improve the accuracy of the method.

This method was never used in this thesis because the results from the earlier method were considered to provide an adequate representation of the behaviour of the gimbal in the simulation environment.

2.4.2 Sensor noise

Sensor noise was added to the model as sets of variances taken from the measurements in Figures 2.14 to 2.17 there the gimbal was held in a fixed configuration.

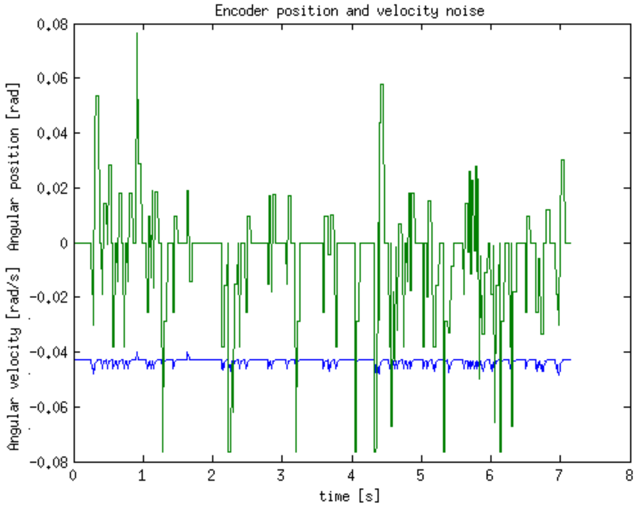


Figure 2.14: *Noise from an encoder measurement after filtering in the gimbal controller. The signal with small variance is the angular position of the encoder and the signal with much bigger variance is the angular velocity which is the filtered derivative of the angular position.*

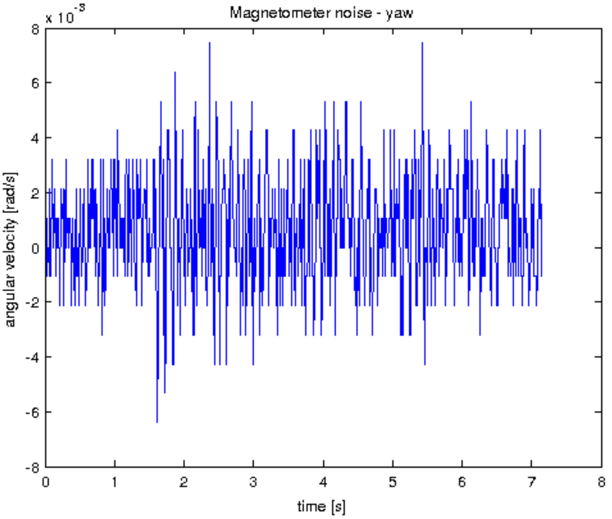


Figure 2.15: *Measurement of the noise in the yaw magnetometer in the IMU when the gimbal was held stationary.*

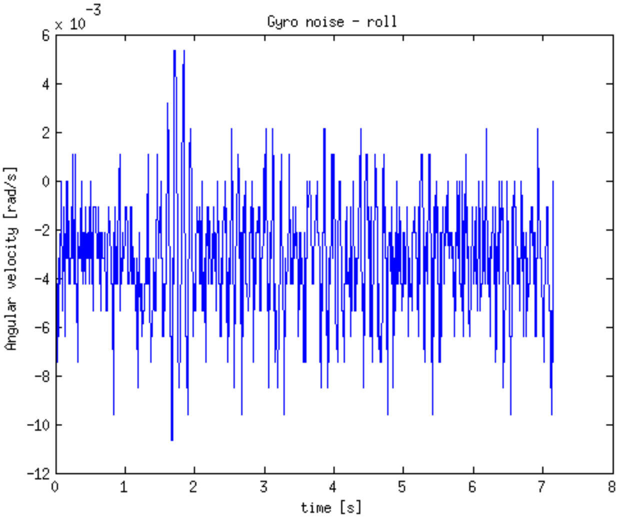


Figure 2.16: *Measurement of the noise in the gyro for the roll axis in the IMU when the gimbal was held stationary.*

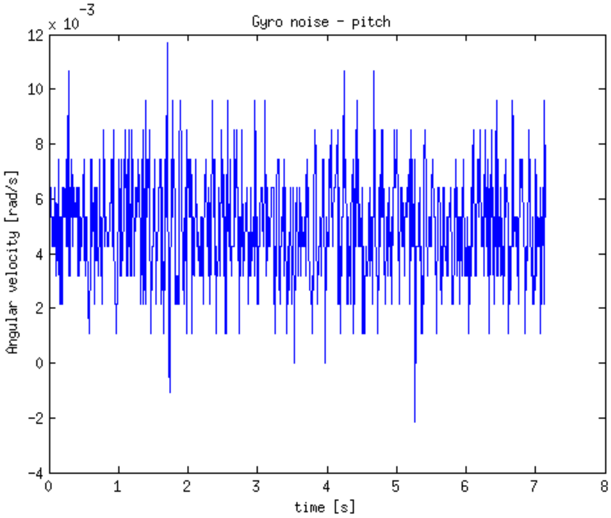


Figure 2.17: *Measurement of the noise in the gyro for the pitch axis in the IMU when the gimbal was held stationary.*

It is shown in figure 2.14 that the angular position noise is mainly due to the last sampled bit in in the signal from the encoder. The real value was somewhere between the value when the least significant bit in the binary signal was 1 or the value when the bit was 0 and the signal jumped between these values. The angular velocity was simply a derivative of the angular position. As a simplification the noise was modelled as white noise with bias and variance according to table 2.2. The noise of the angular position and velocities was also simplified to be viewed as independent.

The largest part of the angular velocity bias was compensated for using the measurements of the IMU. However, a small bias remains after compensation The biases and variances are presented in table 2.2.

The noise on the sensor signals was relatively small due to filtering and bias compensations present in the gimbal controller and can in most cases be disregarded, see section 3.1.

Table 2.2: *Bias and variances of the noise in the encoders and the IMU.*

Signal	**Bias** $[rad(/s)]$	**Variance** $[(rad(/s))^2]$
Encoder angular position	-5.5632e-04	1.2617e-06
Encoder angular velocity	-4.5697e-03	4.5971e-04
Magnetometer yaw	6.7044e-04	4.1017e-06
Gyro roll	-3.1824e-03	5.9804e-06
Gyro pitch	4.9598e-03	4.0390e-06

3

Implementation and control

When the modelling described in chapter 2 was done, a simulation environment was constructed. This simulation environment was then used as a tool in development of the gimbal control structure for the gimbal in the simulation environment and in the gimbal hardware.

3.1 Simulation

The model obtained in section 2.3 was included as in Figure 3.1 in a control structure shown in appendix A. Due to the much too weak motors, the "camera" referred to below can be considered a weightless object mounted on body (3) in Figure 2.1.

3.1.1 Dynamics in Simulink

The dynamic model of the gimbal is set up as a Simulink model (Figure 3.1)

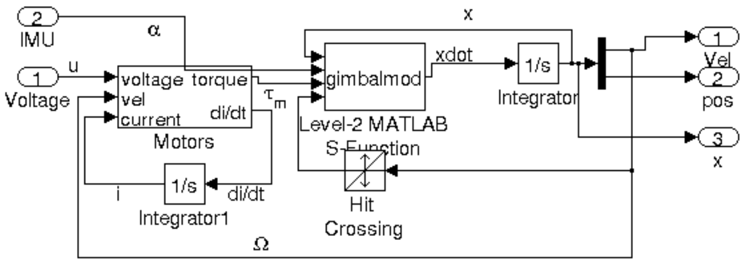


Figure 3.1: Dynamics of the gimbal presented in Simulink.

The block called *gimbalmod* is an *S-function* which contains the differential equations (2.58) and the friction handling. The input is the current state vector $\mathbf{x}$, the generated attitude angles to the ground α, the torque from the actuators τ_m and a check if the angular velocity has reached zero which is used in the friction handling. The output of the function is the derivative state $\dot{\mathbf{x}}$. The differential equations are solved using an integrator and a solver such as ode45.

The *motors* block is set up according to Figure 3.2 and the differential equation of the current i is solved with an integrator.

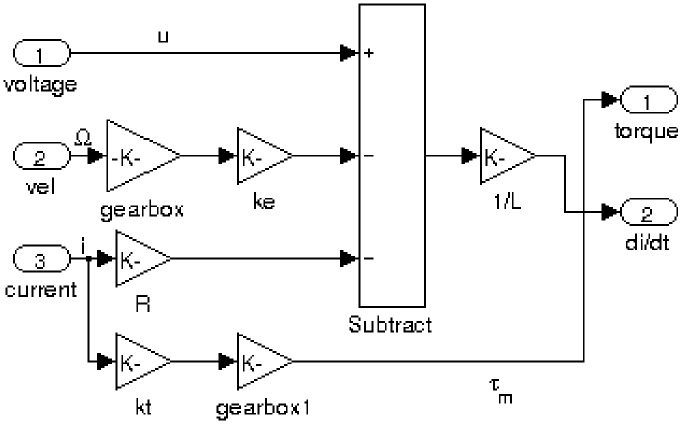


Figure 3.2: *Dynamics of the actuators presented in Simulink.*

3.1.2 Control structure

The control structure shown in Figure A.1 in appendix A is a joint angular velocity and position controller in cascade with an output signal between -1 and 1. The output signal is then multiplied with the battery voltage input to the gimbal motors. This is to get the same behaviour as a by pulse-width modulation (pwm) controlled voltage. There are also input signals for the desired angular positions and velocities in the camera's coordinate frame which are transformed to the joint frames. The last part of the structure in Figure A.1 is a model of how gyros close to the camera would behave when the entire gimbal is disturbed. This structure can be compared to the block diagram in Figure 3.14.

Controllers

The angular position and velocity controllers consists of two ordinary PID algorithms for each axis written as S-functions of the form shown in Algorithm 3.1.

Algorithm 3.1 Pseudo code on the function of the angular velocity controllers. The angular position controllers do not have limited output and does not have any if-statements.

```
Sample($e$);
$v = a + b * e$;

**if** ($v > 1$) **then**
    $u = 1$;
**else if** ($v < -1$) **then**
    $u = -1$;
**else**
    $u = v$;
**end if**
Out($u$);

**if** $|u| \neq 1$ **then**
    $I = I + K\frac{T_s}{T_i}e$;
**end if**
$a = I - K\frac{T_d}{T_s}e$;
$b = K + K\frac{T_d}{T_s} + K\frac{T_s}{T_i}$;
Wait;
```

The variable e in Algorithm 3.1 is the control error, u the output, K the proportional gain constant, I the integral term, T_i the integral time, T_d the derivative time and T_s the sampling interval. a is the additional term of the output which does not depend on the last sample and is pre-calculated in the previous sample and b is the proportional part of the output. This is done, according to [Industriell reglerteknik, ch. 6] to minimize computational delay in the control system. The second if-statement is the condition for integration to avoid reset wind-up. The angular position controllers do not have the if-statements because the output signal is not limited.

The angular velocity controller takes the angular velocity error for each axis as input and outputs three signals between -1 and 1 which is multiplied with the battery voltage. The angular position controller takes the positional error as input and outputs a desired angular velocity to reduce the error. This angular velocity is added to the input to the angular velocity controller.

IMU

In order to disturb the simulation with a motion on the zero frame a block was required that transformed a given angular velocity in zero frame to how the gyroscopes on an IMU close to the camera would measure it. First the components of the angular velocity Φ_0 were transformed to the joints. This can be viewed as if the joints themselves produce the motion $\mathbf{\Omega_\Phi}$ and the zero frame is fixed.

$$\mathbf{\Omega_\Phi} = {}^{\mathbf{joints}}\mathbf{J}_0\mathbf{\Phi_0}, \tag{3.1}$$

where ${}^{\mathbf{joints}}\mathbf{J}_0$ is the Jacobian

$$^{\mathbf{joints}}\mathbf{J}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin(\theta_1) & \cos(\theta_1) \\ \sin(\theta_2) & \cos(\theta_1)\cos(\theta_2) & -\cos(\theta_2)\sin(\theta_1) \end{bmatrix}. \tag{3.2}$$

Then, the angular velocities Ω, which are actually produced by the joints, are added to Ω_Φ. These are then transformed to the camera frame using equation (2.32). The angular velocity that the gyros on the IMU detects is then

$$^3_3\mathbf{\Phi} = \mathbf{J}(\mathbf{\Omega_\Phi} + \mathbf{\Omega}). \tag{3.3}$$

Equation (3.3) is then implemented in the *imusim* block in Figure A.1. The angular position is (as in the hardware implementation) the integral of the angular velocity.

Input

There are a number of inputs to the controller system. The simplest ones are the desired angular positions and velocities in joint space. These require no transformations and are sent directly to the controllers. The angular position control output is, to avoid conflict, simply set to 0 when only angular velocity control is desired. Steering the gimbal to a new angular position using angular velocity control and setting this as the new desired angular position is not implemented in the simulation environment due to the increase in simulation time this brings.

The other input signals are in the camera space according to the modes in section 2.1.3. The difference between the desired angular velocity Φ_d and the, by gyro measured, angular velocity Φ is calculated to determine the required velocity change Φ_e. Φ_e is then in the *invvel* block transformed to joint space using equation (2.38) and added to the current joint angular velocities according to the equation

$$\mathbf{\Omega_d} = \mathbf{J}^{-1}\mathbf{\Phi_e} + \mathbf{\Omega}. \tag{3.4}$$

The desired angular position $\alpha_\mathbf{d}$ is transformed to Θ_d in joint space in the *invpos* block according to section 2.2.2.

Control tuning

The controllers were tuned manually. The nonlinear behaviour of the model made it hard to perform tests (such as transient responses or stable limit cycle tests) in order to determine parameters to be used in tuning methods such as Ziegler-Nichols or Åström-Hägglund [Glad and Ljung, 2006, p. 56]. The angular velocity controller needed to be tuned first and required a proportional gain K and an integral term I with an integral time T_i as input. The parameters were also set differently depending on the sign of the desired angular velocity. The imbalance in the roll axis made the tuning a little bit harder and resulted in an overshoot that could not be reduced by a derivative part. The angular position controllers only needed proportional gains to reach the target which means that it was quite simple to tune a desired "smoothness" on the motion by only lowering the gain K.

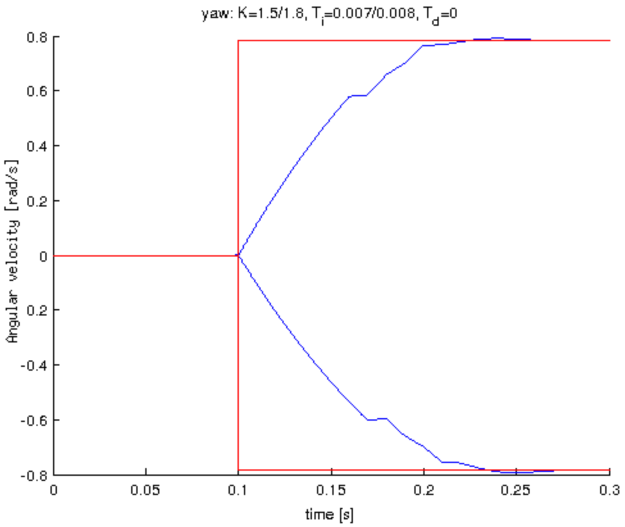


Figure 3.3: *Step responses of the Yaw axis angular velocity control on steps from 0 to $\pi/4$ rad/s in both positive and negative direction. The desired steps are shown as well as the curved responses.*

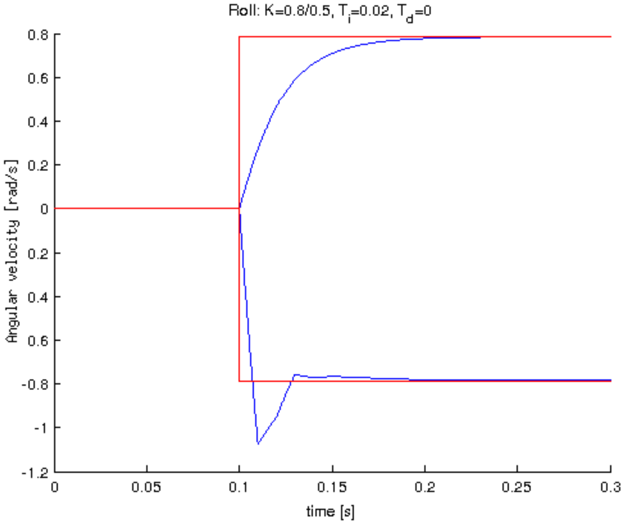


Figure 3.4: *Step responses of the Roll axis angular velocity control on steps from 0 to $\pi/4$ rad/s in both positive and negative direction. The desired steps are shown as well as the curved responses.*

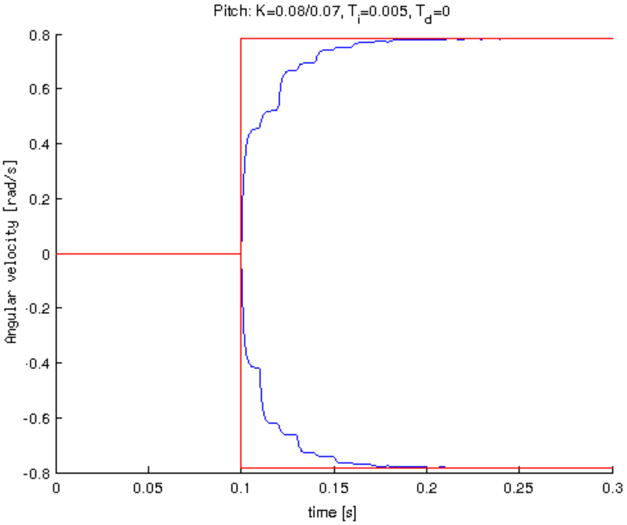


Figure 3.5: *Step responses of the Pitch angular velocity control on steps from 0 to $\pi/4$ rad/s in both positive and negative direction. The desired steps are shown as well as the curved responses.*

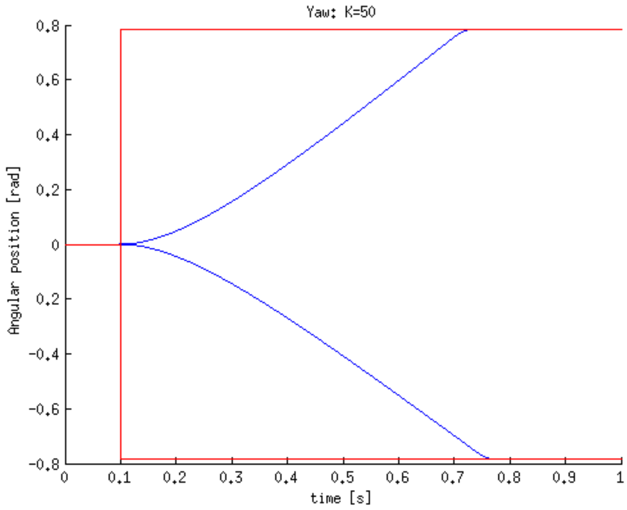


Figure 3.6: *Step responses of the Yaw axis angular position control on a step from 0 to π/4 rad. The desired steps are shown as well as the curved responses.*

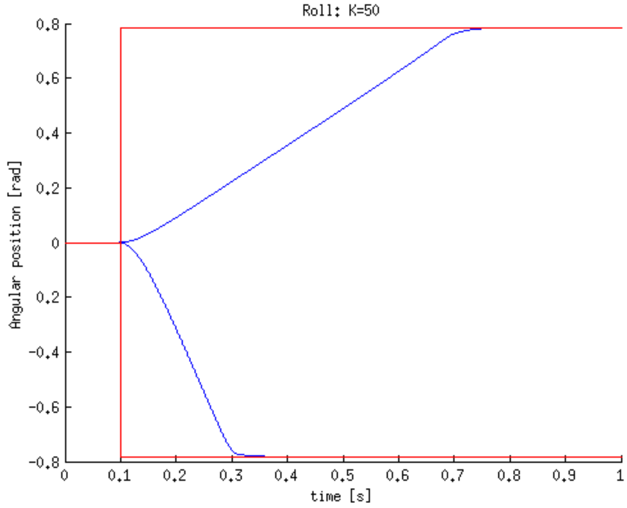


Figure 3.7: *Step responses of the Roll axis angular position control on a step from 0 to π/4 rad. The desired steps are shown as well as the curved responses.*

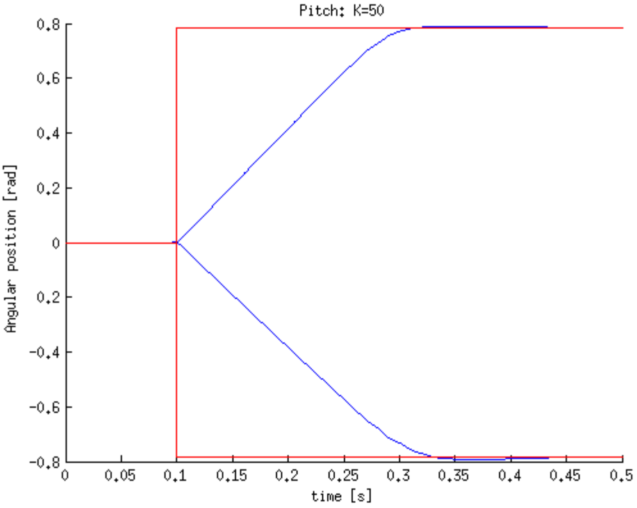


Figure 3.8: *Step responses of the Pitch axis angular position control on a step from 0 to π/4 rad. The desired steps are shown as well as the curved responses.*

3.1.3 Gyro stabilization

A cosine wave on each axis, representing a angular velocity in zero frame, was fed into the *imusim* block described in 3.1.2. In order to counter this distortion the control system had to subtract the, by the gyro, measured angular velocity and position from the reference signals and then transform them to controller inputs. Figures 3.9 to 3.12 show the results for gyro stabilization around the angular positions $\Omega = (0, 0, 0)$.

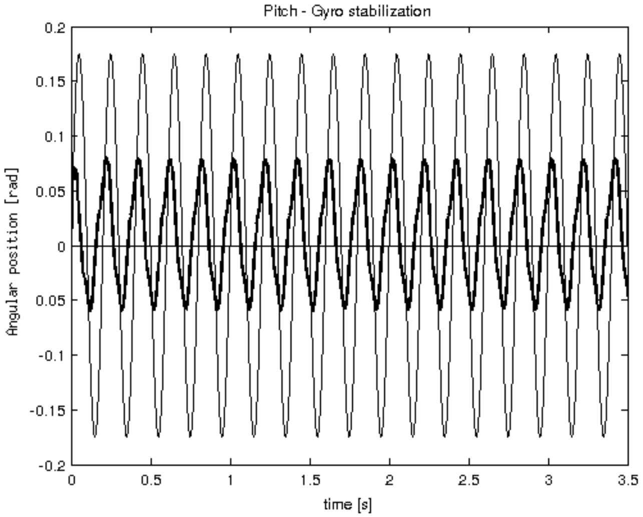


Figure 3.9: *Simulated gyro stabilization of a positional distortion in Pitch in zero frame as a sine wave with a frequency of 5 Hz and an amplitude of 10 degrees. The thin curve is the distortion in zero frame and the bold is the angular position integrated from the IMU signal.*

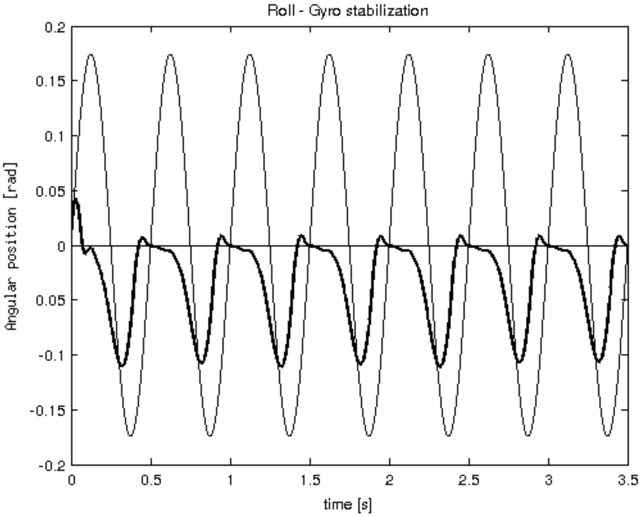


Figure 3.10: *Simulated gyro stabilization of a positional distortion in Roll in zero frame as a sine wave with a frequency of 2 Hz and an amplitude of 10 degrees. The thin curve is the distortion in zero frame and the bold is the angular position integrated from the IMU signal.*

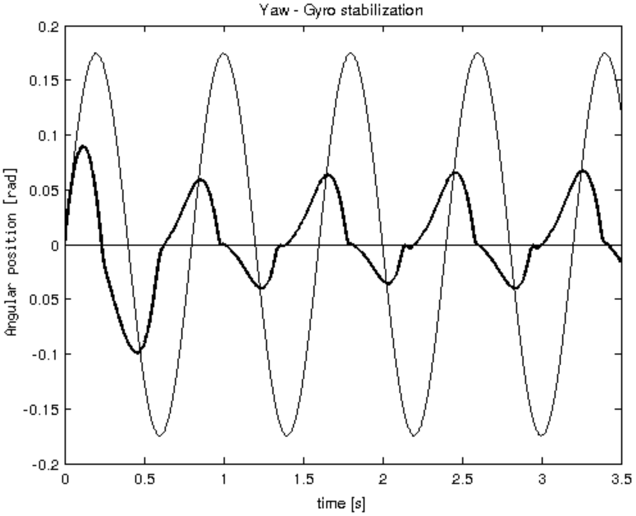


Figure 3.11: *Simulated gyro stabilization of a positional distortion in Yaw in zero frame as a sine wave with a frequency of 1.215 Hz and an amplitude of 10 degrees. The thin curve is the distortion in zero frame and the bold is the angular position integrated from the IMU signal.*

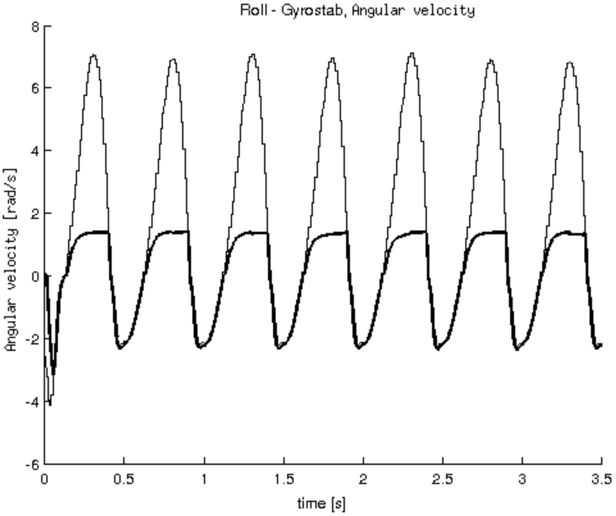


Figure 3.12: *Simulated gyro stabilization of a positional distortion in Roll in zero frame as a sine wave with a frequency of 2 Hz and an amplitude of 10 degrees. Thin curve is the desired joint angular velocity to the to the controller and bold curve is the output angular velocity in the roll joint.*

Figures 3.9 to 3.11 show distortion in frequencies when the gyro stabilization is still "acceptable". The ability of gyro stabilization depends on the angular velocity capability of the joints. The pitch axis has a good stabilization due to its high speed whereas the other two axes have much lower stabilization ability at higher frequencies. This is particularly evident in the unbalanced roll axis (Figure 3.10) where the speed capability is greater when the desired angular velocity is in the negative direction, see Figure 3.12 and Figure 3.4. These simulations show that the motors are much too weak to obtain a good gimbal performance.

Influence by increased noise

The noise measured in section 2.4.2 is relatively small and does not have much impact on the gyro stabilization simulations. If the variances of the noise in the IMU's gyros are multiplied by a factor of $1e^6$ the results are as shown in Figure 3.13. The drift in the angular position is because the control system is entirely relying on the accuracy of the IMU gyro measurements and has no other reference. Increased noise in the joint encoders does not have any significant impact on the gyro stabilization. This is because the control system only uses the control error from the IMU. This would not be the case if the IMU gyros and the encoders were read at different sample rates or if the joints are controlled directly, for instance during initiations and calibrations.

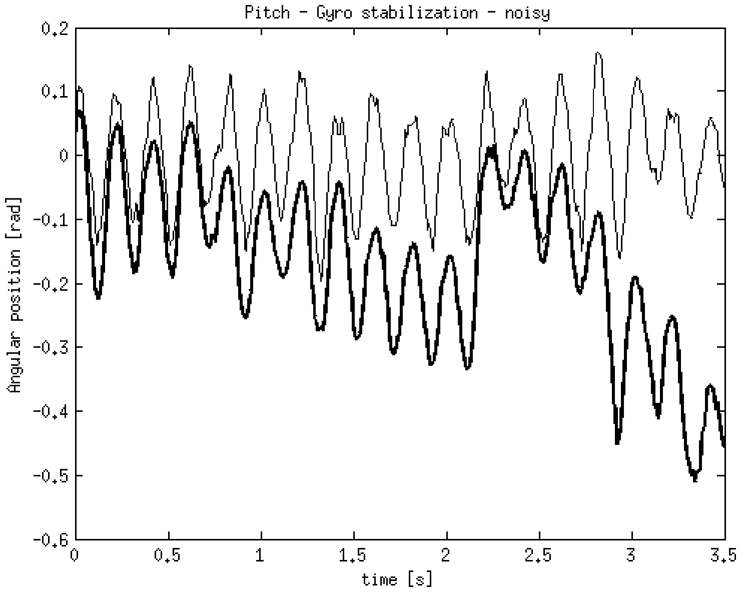


Figure 3.13: *Gyro stabilization test on pitch where the variance of the measurement noise in the IMU gyros was multiplied by a factor of* $1e^6$ *(all axes). The thin curve is the angular position, as integrated from the IMU signals, and the bold curve is the angular position, as measured without any noise.*

3.2 Hardware Implementation

The control implementation on the gimbal controller card is written in C-code and uses the same kinematic functions (section 2.2.2) as the control structure in the simulation environment. The major challenge in hardware programming was to implement the encoders and the gyros and accelerometers on the IMU correctly and use their data to derive the different angles and angular velocities needed. The controllers were then tuned with the simulation results in section 3.1.2 as a starting reference.

3.2.1 Numerical representation

To eliminate the need for floating-point calculations, floating-point numbers are represented as Texas Instruments (TI) IQ24 numbers:

$$\mathrm{IQ24}(number) = number \times 2^{24}. \tag{3.5}$$

The IQ number can be treated as any integer or used by TI-special functions such as correct multiplications and trigonometric functions. The scale 2^{24} is a tradeoff between numerical accuracy and the range of possible values, in this case range ± 128 and accuracy $6 \cdot 10^{-8}$. The scaling factor can vary between 2^1 and 2^{30} where a larger scaling factor means higher accuracy but a smaller range.

3.2.2 Code structure

Figure 3.14 shows a block diagram over the control structure in the hardware implementation. It can be compared to Figure A.1. A flow chart of the program code is shown in Figure 3.15. The program contains the following parts:

Input reading

The gimbal controller has the ability to read input data from a computer via serial communication (USB converted to UART). The input data consists of strings which are interpreted by the controller. The data is usually a new desired angle α_d or angular velocity Φ_d in the camera space but it has also been used for setting desired joint positions Θ_d or joint angular velocities Ω_d and new PID controller parameters.

Update of joint states

The rotary encoder on gimbal axis i measures its current angular position and sends it as a 10 bit binary number to the control unit. The control unit scales the number to an angle θ_i in radians (in IQ24 presentation, section 3.2.1). θ_i, which originally lies in the range $0...2\pi$, is also converted to lie in the range $-\pi...\pi$. This means $\theta_i = 0$ on the gimbal is 180 degrees from angle 0 on the encoder.

If the angular distance between two measurements at sample k and $k-1$ is large enough it is used to approximate the angular velocity ω_i using the sample frequency $F_s = 100$ Hz.

$$\omega_i(k) = (\theta_i(k) - \theta_i(k-1))\,F_s \tag{3.6}$$

The angles θ_i and the angular velocities ω_i are filtered using a simple exponential moving average:

$$\theta_i = \frac{1}{2}\theta_i + \frac{1}{2}\theta_{i,raw} \tag{3.7}$$

and

$$\omega_i = \frac{3}{4}\omega_i + \frac{1}{4}\omega_{i,raw}, \tag{3.8}$$

where $\theta_{i,raw}$ and $\omega_{i,raw}$ are the unfiltered angular position and velocity in the current sample. The smoothing factors $a = \frac{1}{2}$ and $a = \frac{1}{4}$ were only roughly tuned until an acceptable balance between accuracy and smoothness was believed to be found.

Update of IMU states

The gyros on the IMU close to the camera mounting point measures the current yaw, roll and pitch angular velocities $\Phi = \left[\phi_{yaw}, \phi_{roll}, \phi_{pitch}\right]^T$ of body 3 around its own axes. It is set to measure in the range of 0... ± 2000 °/s and sends the measured angular velocities to the control unit as 15 bit binary numbers which are then scaled to radians in the IQ24 representation (section 3.2.1). The measurements are calibrated by subtracting a bias measured at the beginning of each gimbal session.

Φ is used to calculate the current angular position relative to the ground α. As mentioned in section 2.1.2 attitude calculations using the measured angular velocities suffer from angular drift due to accumulated measurement errors. A simple approximate integration of Φ using the sample frequency F_s in this case also caused problem with angular drift. This was however solved by the use of a *complementary filter* already written by IA. The details of this filter were not studied enough to be presented in this thesis but basic function is to integrate the by gyro measured angular velocity to an attitude which is filtered using the translational acceleration Γ. This acceleration is measured by accelerometers on the IMU and are sent to the control unit in a similar way as the angular velocities.

Inverse kinematics

The desired camera angular position α_d or the desired camera angular velocity Φ_d is transformed to reference signals to the joint controllers in the same way as in section 3.1.2 using equations in section 2.2.2 and equation (3.4).

Update of control signals

The code for the PID-controllers was written in according to Algorithm 3.1. The accumulated integral term I, the additional term a and the proportional part p are stored in a static structure to be used in the next sample. In total there are 6 structures, one for each joint state.

As in the simulation environment in section 3.1.2 the angular position controller takes the desired angular positions from the inverse kinematics as reference signals and outputs the desired angular velocity for each axis. These angular velocities are added to the desired angular velocities Ω_d from the inverse kinematics which become reference signals to the angular velocity controller. The output from the angular velocity controller is a number between 0 and ±1 in the IQ24 representation. This number is rescaled to an integer between 0 and ±1000 which is the permille (‰) of the PWM signal to the motor output ports, or the permille of the maximum voltage to the motors. The voltages create a torque on each motor according to section 2.3.2, which sets the gimbal in motion.

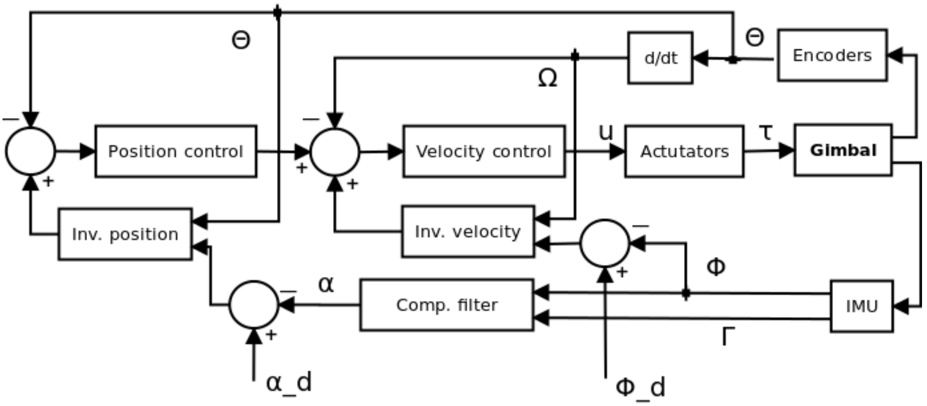


__Figure 3.14:__ Block diagram of the control structure in the gimbal hardware implementation.

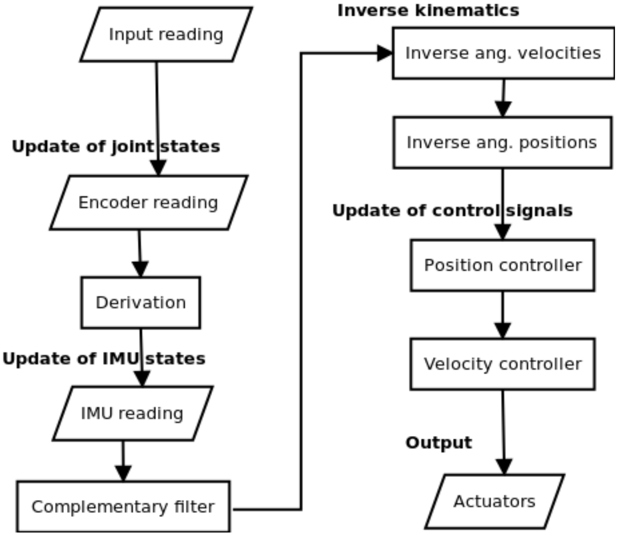


Figure 3.15: *Flow chart of one loop in the gimbal control code after initiation and calibration.*

3.2.3 Control tuning

The PID-controllers of the gimbal were tuned by first applying the PID parameters which were tuned in the simulations described in section 3.1.2. These were then adjusted to attain acceptable performance in the real application.

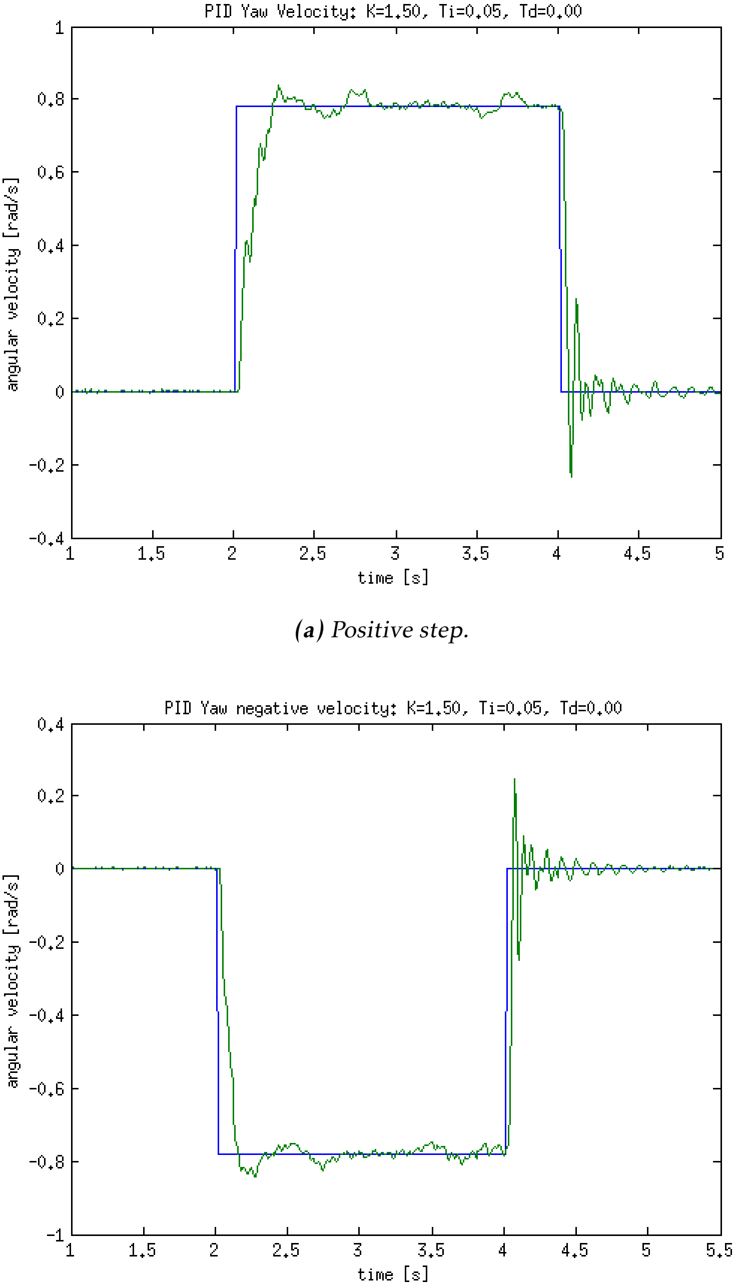
Angular velocity

The PID-parameters from the simulations (section 3.1.2) yielded unstable results when used in the real gimbal control. The approach to tune the controller parameters on each axis was to keep the proportional gains K from the simulations and adjust the integral time T_i until performance was deemed acceptable. This resulted in the responses in Figures 3.16 to 3.18 and the parameters in table 3.1. Two of the gain parameters K were also slightly adjusted. The results of the angular velocity control became somewhat oscillative but still but were still considered acceptable with a balance between rise time, oscillations and accuracy. Worth mentioning is the difference in behaviour between the first and second transients in Figure 3.16 where the angular velocity in yaw axis steps back 0 rad/s are much more oscillative than the first steps. A possible explanation is the nonlinear behaviour in the Yaw axis. The gimbal turns to around 180 degrees during both motions and stops in the same angular position where the behaviour of the joint is different from the behaviour when the yaw axis is in angular position 0 degrees. Another set of controller parameters would be more appropriate in this angular position. It is also worth to notice that the pitch signal in Figure 3.18 is much noisier when the gimbal is given a angular velocity than when the angular velocity is zero. A possible explanation is that the motion itself induces small vibrations in the joint and in the body.

Table 3.1: *Angular velocity PI parameters in the real gimbal control, parameters from simulations in parentheses.*

Axis (direction)	**K**	$\mathbf{T_i}$
Yaw (+)	1.5(1.5)	0.05(0.007)
Yaw (-)	1.5(1.8)	0.05(0.008)
Roll (+)	0.8(0.8)	0.05(0.02)
Roll (-)	0.4(0.5)	0.05(0.02)
Pitch (+)	0.08(0.08)	0.01(0.005)
Pitch (-)	0.07(0.07)	0.01(0.005)

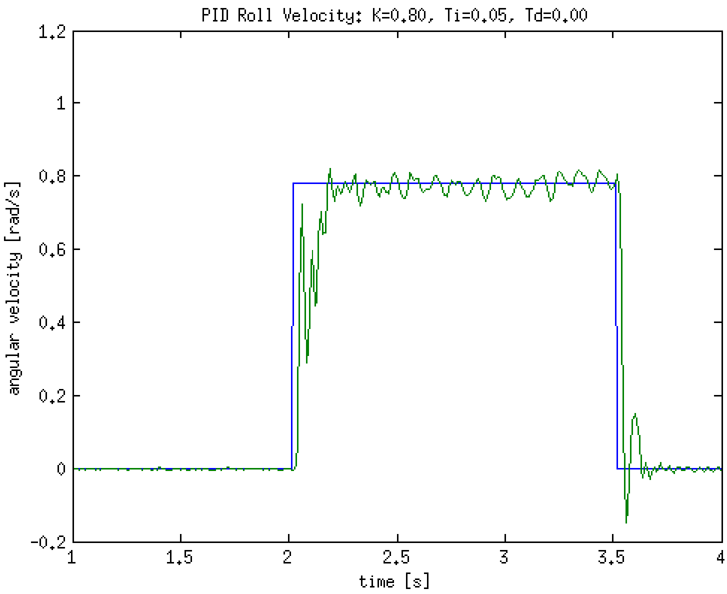
The relatively large difference between the T_i parameters from the real control and from the simulations can be explained by looking at the angular velocity rise times in the figures with the adjusted friction parameters in section (2.4.1). The rise time of the simulation in Figure 2.5 is for instance relatively close to zero whereas the actual rise time is longer. In tuning methods, such as Ziegler-Nichols or Åström-Hägglund [Glad and Ljung, 2006, p. 56] where step responses are used to get reasonable values of the control parameters, the integral time T_i is proportional to the rise time which means that a longer rise time demands a longer integral time. If T_i is too small, it will create a too large integral term in the control (see Algorithm 3.1).



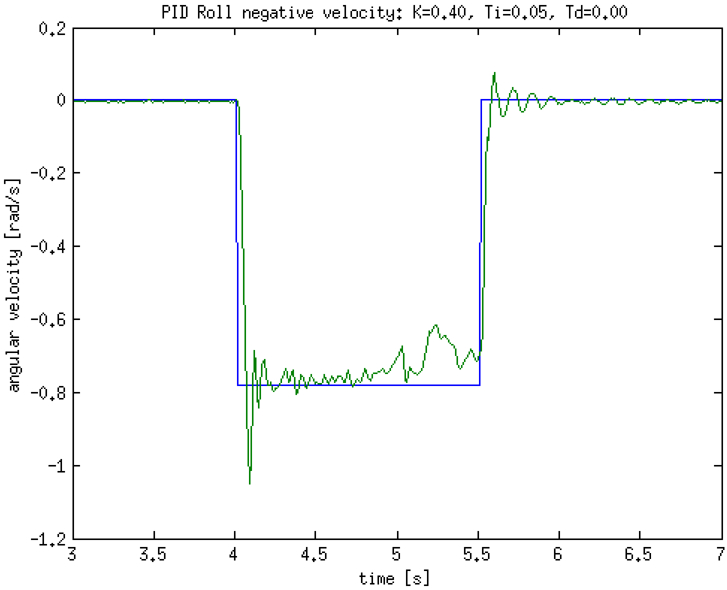
***(a)** Positive step.*

***(b)** Negative step.*

***Figure 3.16:** Step responses of the angular velocity in the yaw axis with tuned PID parmeters. Figure a) shows a step from 0 to π/4 rad and b) a step from 0 to −π/4 rad.*

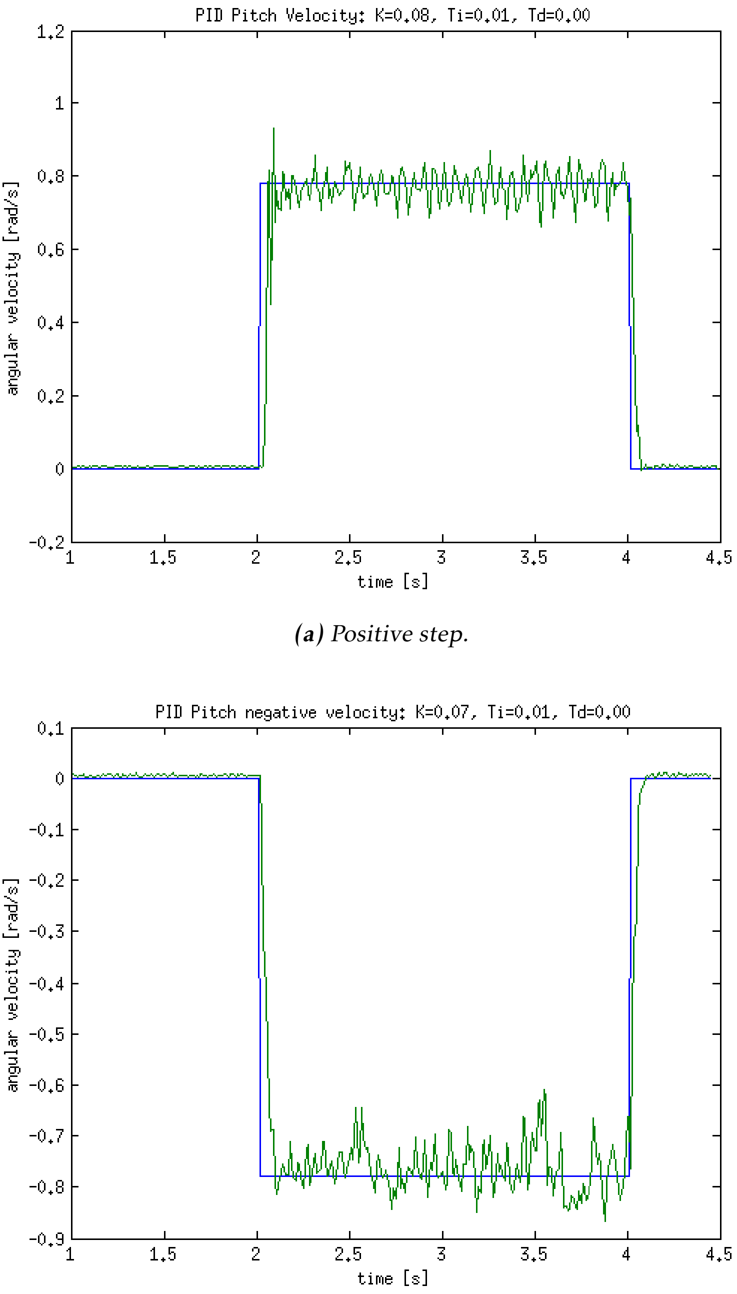


*(**a**) Positive step.*



*(**b**) Negative step.*

Figure 3.17: *Step responses of the angular velocity in the roll axis with tuned PID parmeters. Figure a) shows a step from 0 to $\pi/4$ rad and b) a step from 0 to $-\pi/4$ rad.*



***(a)** Positive step.*

***(b)** Negative step.*

***Figure 3.18:** Step responses of the angular velocity in the pitch axis with tuned PID parmeters. Figure a) shows a step from 0 to $\pi/4$ rad and b) a step from 0 to $-\pi/4$ rad.*

Angular position

The gains K for the angular positions of the gimbals axes were adjusted from 50 to 40 in order to obtain smoother results. Figures 3.19 to 3.21 show the desired angular position and with IMU estimated angular position of the camera. Figure 3.21b also shows the desired and measured angular position in the pitch joint to illustrate the nature of the angular position estimates. The overshoots in Figure 3.21b occur because the complementary filter mentioned in 3.2.2. The filter uses both integration of the angular velocities from the gyros and the direction of the translational acceleration from the accelerometers. The angular velocity part is very sensitive to changes but the acceleration part reacts much slower so it takes a while before the estimates are adjusted. The positional control follows this adjustment.

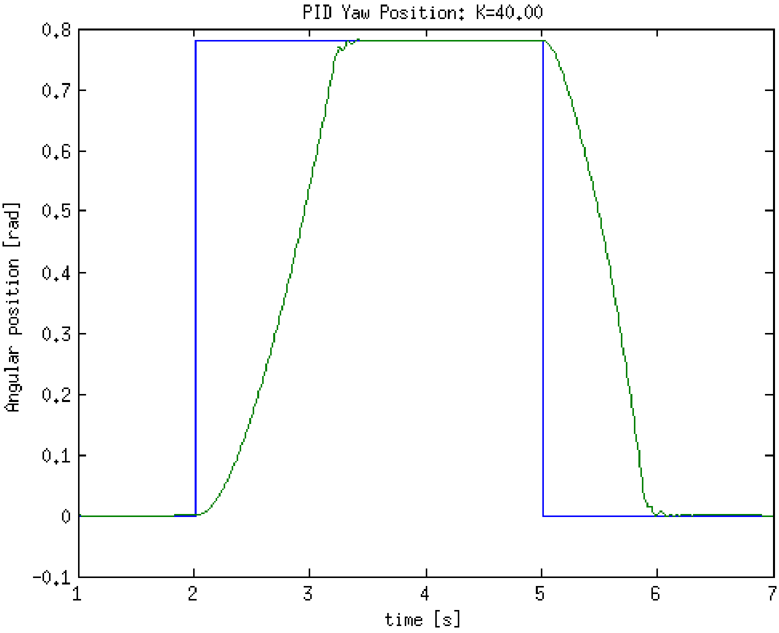
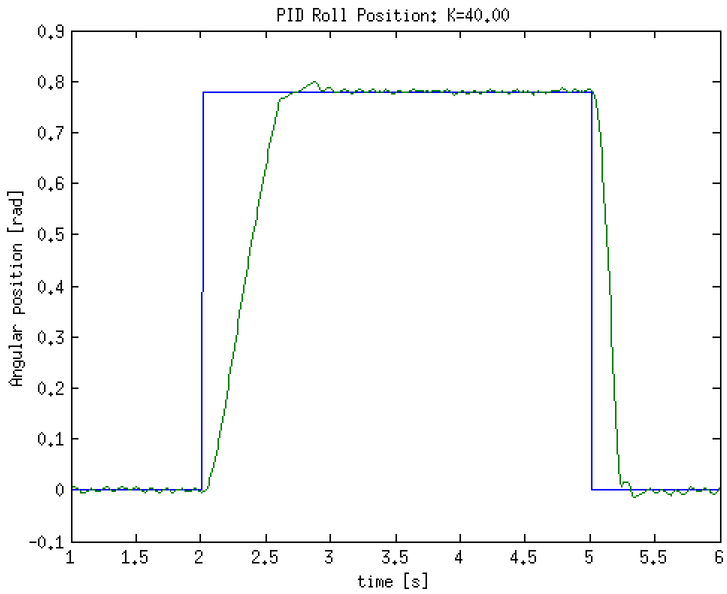
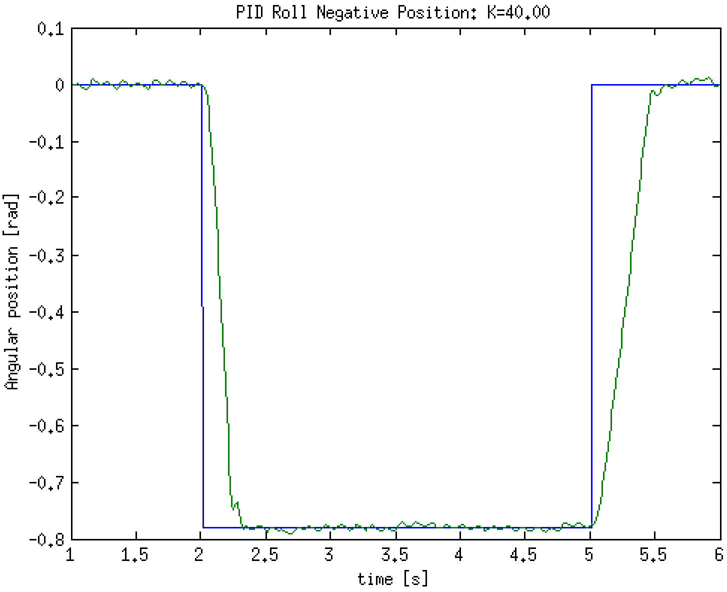


Figure 3.19: *Step response of the angular position in the yaw axis with tuned PID parmeters. Step from 0 to π/4 rad.*

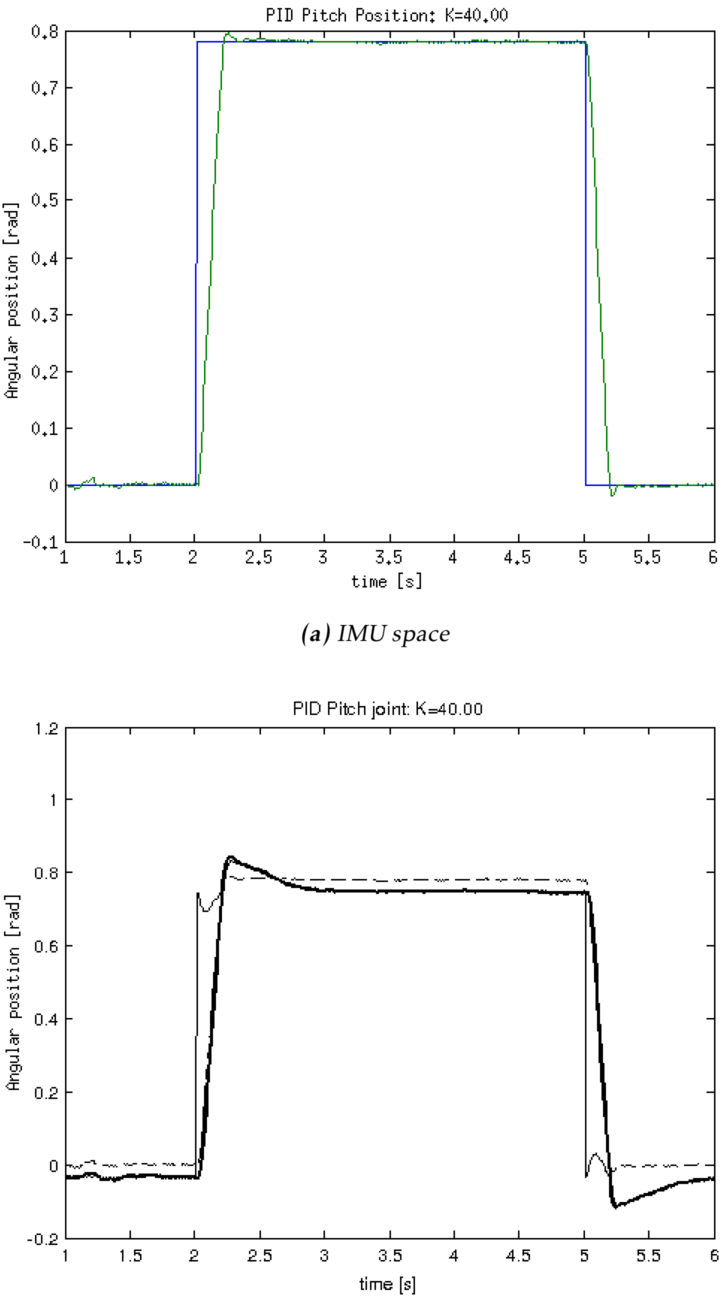


(a) Positive step.



(b) Negative step.

Figure 3.20: *Step responses of the angular position in the roll axis with tuned PID parameters. (a) shows a step from 0 to $\pi/4$ rad and (b) a step in the negative direction to illustrate the significant rise time difference between the directions.*



***(a)** IMU space*

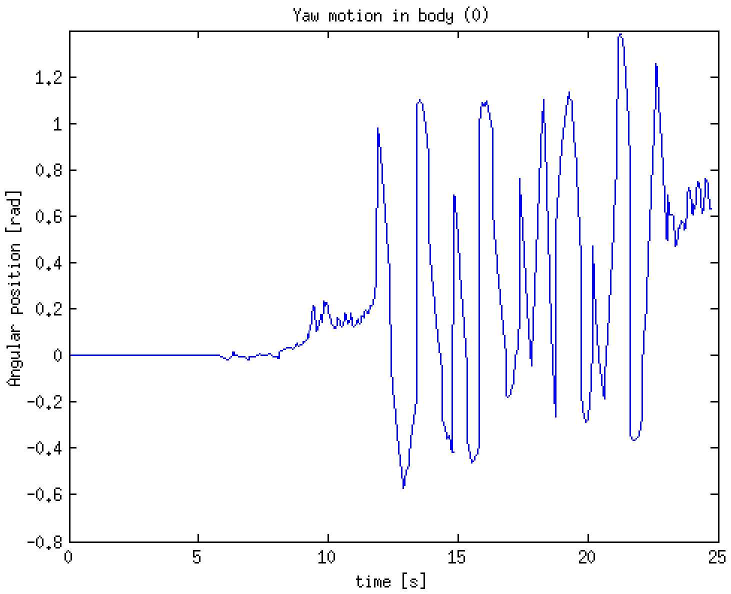
***(b)** Joint space.*

***Figure 3.21:** Step responses of the angular position in the pitch axis with tuned PID parameters. (a) shows the angular position estimated in the complementary filter (Section 3.2.2). (b) Shows the angular position in joint space (bold curve) and the desired joint angular position (solid, thin curve). The dashed curve is the angular position in IMU space, same as in (a).*

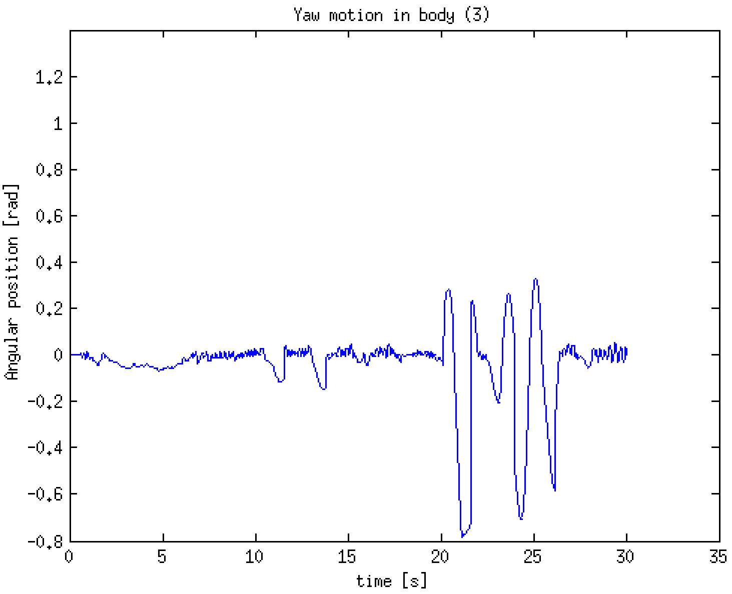
3.2.4 Gyro stabilization

Tests were done to determine the gimbal controller's ability to perform active stabilization of the camera. This was done by comparing a measured motion in body frame (0) to the motion in body frame (3) (see Figures 2.1 and 2.2). The measurements in frame (3) were done with the IMU connected to the gimbal control unit and the measurements in frame (0) were done with a second IMU connected to a separate control unit. The two control units were not synchronized so the same disturbance motion was recorded at different times by the two units. This time difference was about 1.5 seconds. There was also an angular drift in the yaw axis of the IMU on body (0) but, as testing was mainly concerned with the local amplitudes and frequencies and not the angular position, this did not cause a problem.

Figures 3.22 to 3.24 show the results from tests where the entire gimbal was given a motion in one axis at a time. The gimbal was held by hand in body (0) and was turned in the yaw axis or rocked in the roll and pitch axis of that body. Figure 3.25 shows the results when the gimbal was moved in all three axes. All these measurement started in the IMU angular position $\alpha = (0, 0, 0)^T$.

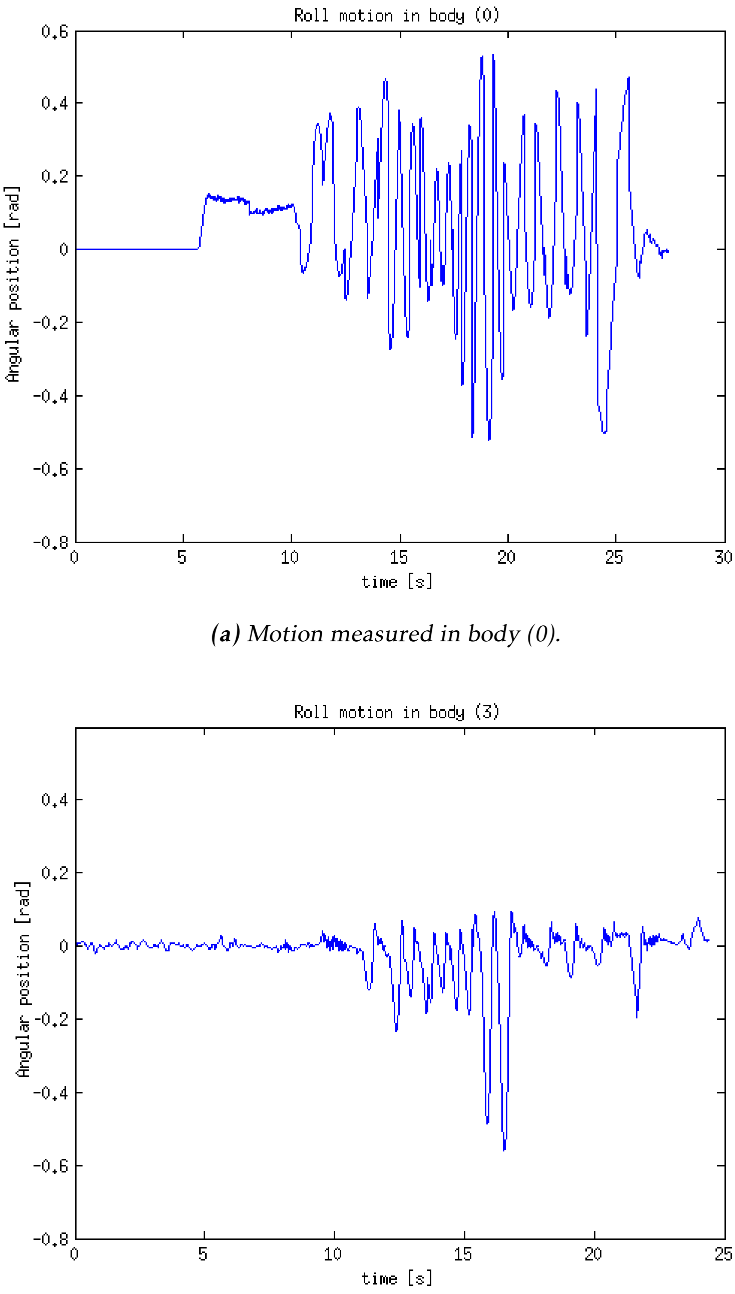


*(**a**) Motion measured in body (0).*



*(**b**) Motion measured in body (3).*

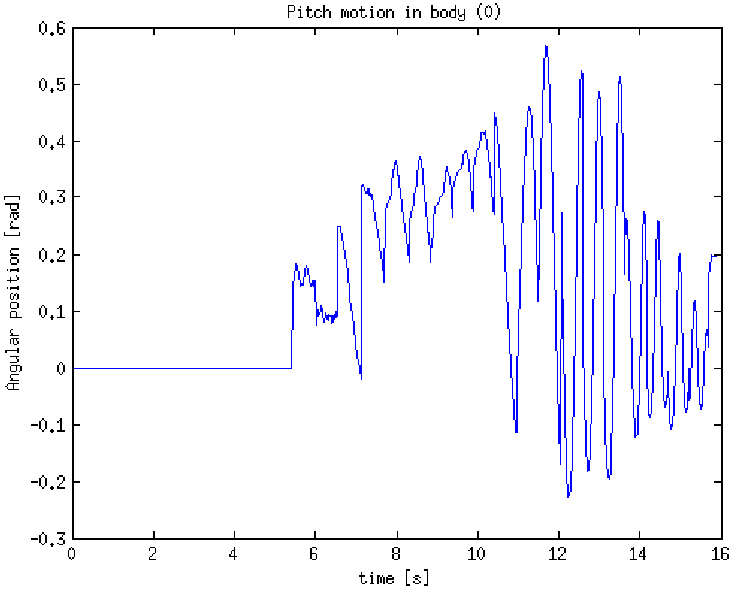
Figure 3.22: *The result from a test of the gyro stabilization in the yaw axis. (a) shows the measurement of the motion of the entire gimbal measured with a IMU attached to body (0) (Figure 2.1). (b) shows the motion of body (3) measured with the IMU there. The measurements were not synchronized, the upper measurement started about 1.5 s before the lower.*



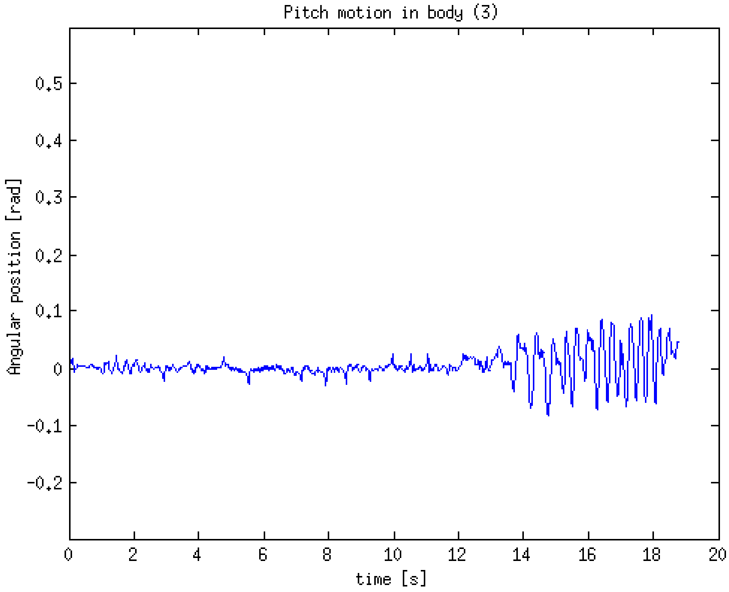
***(a)** Motion measured in body (0).*

***(b)** Motion measured in body (3).*

***Figure 3.23:** The result from a test of the gyro stabilization in the roll axis. (a) shows the measurement of the motion of the entire gimbal measured with a IMU attached to body (0) (Figure 2.1). (b) shows the motion of body (3) measured with the IMU there. The measurements were not synchronized, the upper measurement started about 1.5 s before the lower.*

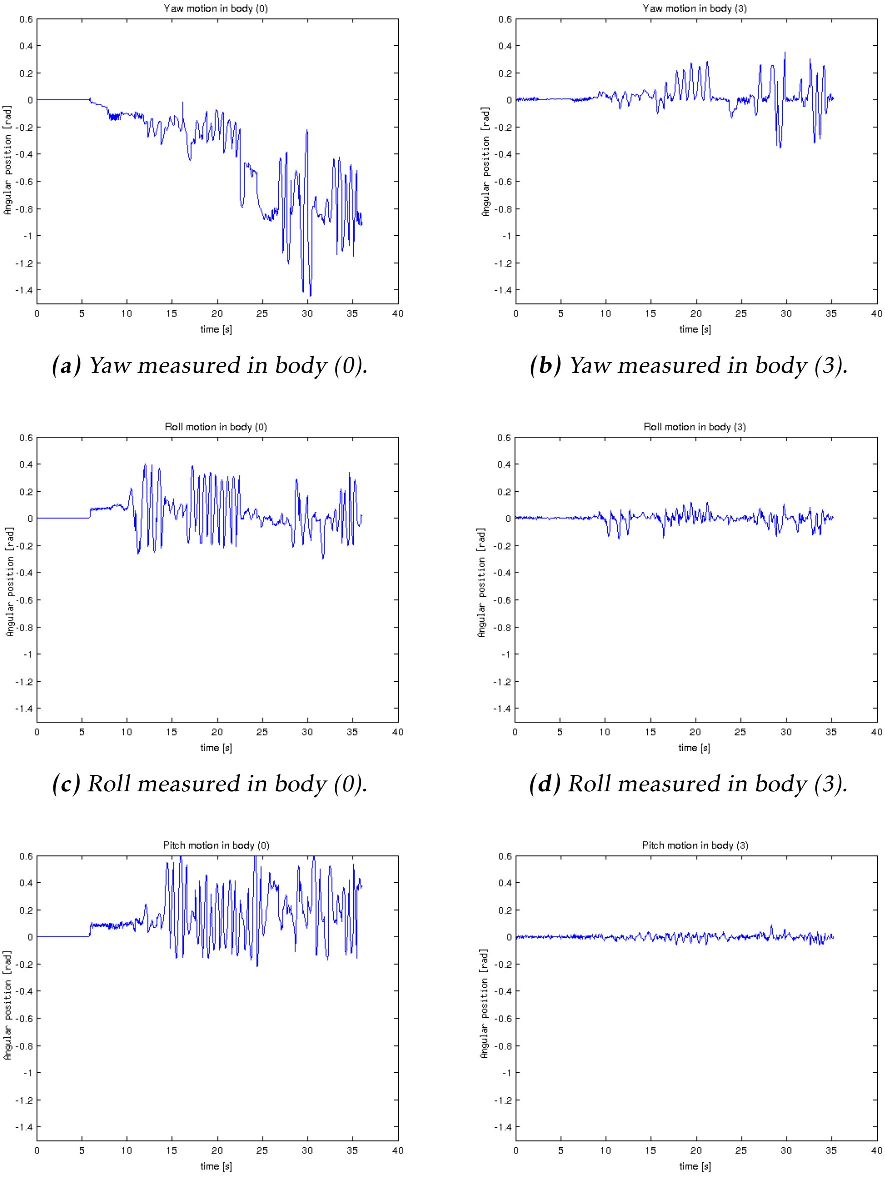


(a) Motion measured in body (0).



(b) Motion measured in body (3).

Figure 3.24: *The result from a test of the gyro stabilization in the pitch axis. (a) shows the measurement of the motion of the entire gimbal measured with a IMU attached to body (0) (Figure 2.1). (b) shows the motion of body (3) measured with the IMU there. The measurements were not synchronized, the upper measurement started about 1.5 s before the lower.*



(a) *Yaw measured in body (0).*

(b) *Yaw measured in body (3).*

(c) *Roll measured in body (0).*

(d) *Roll measured in body (3).*

(e) *Pitch measured in body (0).*

(f) *Pitch measured in body (3).*

Figure 3.25: *The results from a test of the gyro stabilization in all three axes where the gimbal was rocked in all three directions. The figures to the left show the measurements of the motion of the entire gimbal measured with a IMU attached to body (0) (Figure 2.1). The figures to the right show the motion of body (3) measured with the IMU there. The measurements were not synchronized, the measurements to the left started about 1.5 s before the ones to the right.*

The results are similar to the results from the simulations in section 3.1.3. The ability to stabilize differs between the axes. The pitch axis has a very good stabilization while the ability in the roll and yaw axis is poorer when the frequency is increased to somewhere around 1 to 2 Hz. This, again illustrates the need for much stronger actuators to obtain the needed angular velocities.

4 Conclusions and discussion

This final chapter contains some conclusions drawn in this thesis, thoughts on what could have been done differently, and suggestions for future work.

4.1 About the modelling

The model of the gimbal derived in chapter 2 in this thesis does not exactly match the behaviour of the real gimbal but is, for instance, sufficient for development of control structures. One of the hardest parts of this work was to verify that the transformations between the different coordinate frames were correct. This was done by moving the joints of a crude little physical model of the gimbal and performing the same motions in the simulation environment. It was important that this was done before the hardware implementation in order to avoid damages to the gimbal due to unexpected motions.

The method using Lagrange equations with link transformation matrices in section 2.3.1 provides a way to calculate the dynamic model using fairly simple programming in Matlab. However the calculation did take a lot of time, especially if simplification algorithms were used in the process. The expressions did also became very cumbersome. They was improved slightly by transferring them to Mathematica and using the simplification algorithms included in the program. The results would probably be even better, in terms of faster computations and more compact expressions, if all calculations had been done using Mathematica or other tools for symbolic calculations.

4.1.1 What could had been done differently

As mentioned in the end of section 2.3.2 the actuators could be treated as separate bodies in the model, this to to allow testing of different motors and gearboxes without the need to calculate an entirely new model. This would also have been useful to find more powerful actuators and also test how the performance was affected by their increased size and mass..

Parameters of the gimbal were taken from drawings in CAD software. The same kind of software could be used to simulate a multi body systems and even develop control systems around them. However, this possibility was not studied in this thesis. One argument for the mathematical modelling method used in this thesis is that it is, in principle independent of expensive software (even tough the dimensions and masses could be less accurate if measured by hand and the calculations would be tedious without proper linear algebra software). The accuracy of the model would probably increase with more powerful tools for identification of unknown parameters, mentioned in section 2.4. Simulations could also be done in any programming language, provided differential equation solvers were available.

4.1.2 Suggested future work

As explained in section 2.4.1 model accuracy can be improved by more sophisticated identification methods. This would create a model accurate enough to be used in model based control methods or as residuals in diagnostics systems. Implementing the model in hardware would probably require a more powerful processor and better simplification algorithms to get less complicated expressions.

Modelling methods where powerful mechanics software is used should also be evaluated.

4.2 About the control structure

The control system of the gimbal was only developed to perform around one point ($\alpha = (0, 0, 0)$) and did not get more advanced controllers than a P-controller for angular position and a PI-controller for angular velocity of each axis. One big advantage of PI(D)-controllers is that they are relatively simple to work with for someone with limited prior knowledge of control theory. The hardest part of the control design, apart from the work of writing the code, was to derive and verify the transformations between camera and joint frames. This needed to be done with great care to avoid unexpected motions that could potentially damage the gimbal. The implementation was successful and no unexpected motions were observed.

4.2.1 What could have been done differently

The sample rate in the control system was set quite low (100 Hz). Higher sample rates would probably give some performance improvement. The low sample rate was mainly to avoid data loss in the serial communication between the gimbal and a computer.

4.2.2 Suggested future work

The performance would probably be improved if linearization was used. It was suggested during the thesis to have different sets of control parameters for different configurations of the gimbal, however, this was not implemented. Changes in the mass moments of inertia resulting from camera reorientation would lead to varying performance at different angular positions of the camera. Linearization would be done by having a number of angles as linearization points and having a catalogue of control parameters which become active in the vicinity of their assigned linearization points. Interpolation between the points has also been discussed.

Other types of controllers, such as lead-lag or LQ-controllers, should be evaluated. With better performance of the actuators and therefore higher accelerations, coupling effects between the joint states would become a bigger problem. The results from Seong et al. [2006] suggests that a LQG/LTR control (Linear Quadratic Gaussian control with Loop Transfer Recovery) would perform better in terms of decoupling.

With more powerful microprocessors it would be interesting to use model based control, as suggested in 4.1.2. One other possibility would also be to send a model based feed-forward signal to the platform with information about the gimbal motion so the platform control could counteract platform motion due to motion of the gimbal. This would reduce the risk of unstable oscillations induced by the control systems. The model would be a Newton-Euler dynamic model of the reaction forces in the yaw joint.

A

Simulink control structure

Figure A.1 shows the control structure in the Simulink environment. In broad strokes, it can be interpreted in the following way: The blocks in the upper part of the structure pertain to states in joint coordinate space; control of joint angular position and velocity in interaction with the mathematical model of the gimbal. The "invpos" and "invel" blocks in the middle of the structure are the transformations between IMU coordinate space and joint space. The blocks in the lower right corner model the behaviour of the gyros (and the magnetometer) in the IMU and external manipulation of the gimbal. More detailed descriptions of the control structure can be found in section 3.1.2.

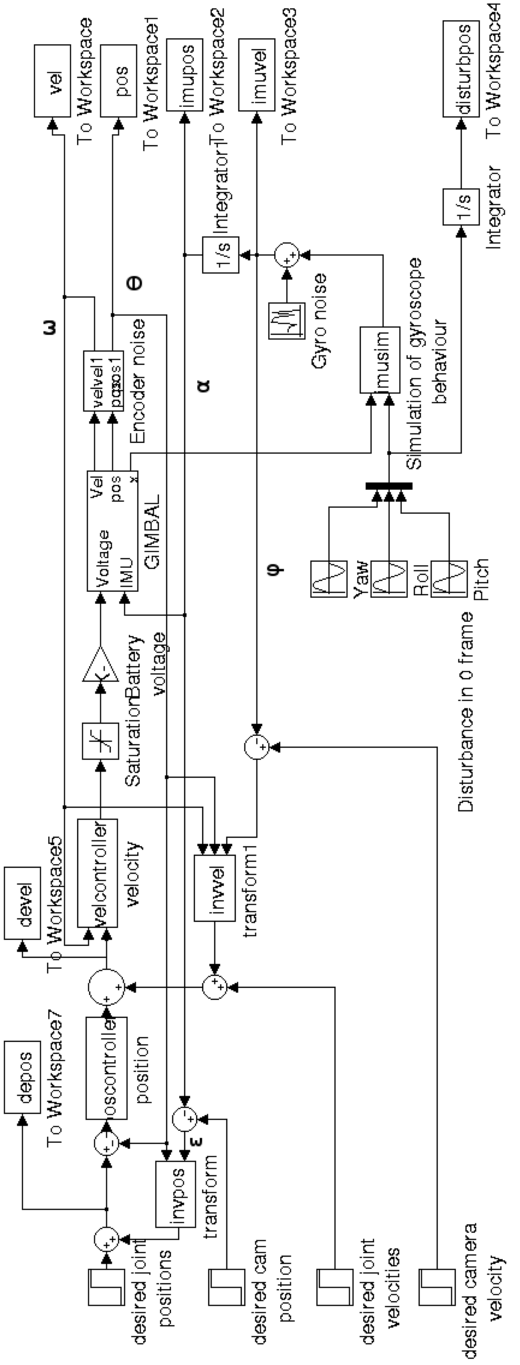


Figure A.1: *Control structure in Simulink.*

Bibliography

T Glad and L Ljung. *Reglerteknik: grundläggande teori*. Studentlitteratur, 2006. Cited on pages 39 and 50.

Industriell reglerteknik. Kurskompendium. Reglerteknik, Institutionen för systemteknik, Linköpings Universitet, 581 83 Linköping, 2009. Cited on page 37.

R N Jazar. *Theory of Applied Robotics: Kinematics, Dynamics, and Control*. Springer, 2007. Cited on pages 11, 12, 15, 18, 19, 20, 21, and 23.

Y S Kwon, H Y Hwang, and Y S Choi. Stabilization loop design on direct drive gimbaled platform with low stiffness and heavy inertia. In *International Conference on Control, Automation and Systems*, 2007. Cited on pages 4 and 5.

N Mohan, T M Undeland, and W P Robbins. *Power Electronics, Converters, Applications, and design*. Wiley, 2003. Cited on page 24.

T Parveen. Dynamic analysis of 8-joint gimbal system executing 4 dof motion. In *International Bhurban Conference on Applied Sciences & Technology*, 2009. Cited on page 4.

K J Seong, H G Kang, B Y Yeo, and H P Lee. The stabilization loop design for a two-axis gimbal system using LQG/LTR controller. In *SICE-ICASE International Joint Conference*, 2006. Cited on pages 5 and 65.

K J Seong, H G Kang, B Y Yeo, and H P Lee. The digital anti-windup LQG/LTR controller design for a two axis gimbal system. In *International Conference on Control, Automation and Systems*, 2007. Cited on page 5.

Y B Shtessel. Sliding mode stabilization of three axis inertial platform. In *System Theory, 1994., Proceedings of the 26th Southeastern Symposium*, 1999. Cited on page 5.

P Skoglar. Modelling and control of IR/EO-gimbal for UAV surveillance applications. master's thesis lith-isy-ex-3258-2002. Master's thesis, Department of Electrical Engineering, Linköpings Universitet, 2002. Cited on page 4.

B J Smith, W J Schrenk, W B Gass, and Y B Shtessel. Sliding mode control in a two axis gimbal system. In *Aerospace Conference, 1999. Proceedings. 1999 IEEE*, 1999. Cited on page 5.