Gyro Stabilization of a Positioning Unit

Wilhelm Andrén

Ella Hjertberg

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6082
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2019

Abstract

Mounting cameras on motorized objects has become possible in greater extent due to the emerging camera technology during the past few decades. This application could prove useful in several areas such as search and rescue operations, surveillance or even news monitoring. One hardship that this brings is the difficulty of keeping the camera unit stable while its setting is not. In the situations mentioned above, it would be desired to keep the camera unit in the same attitude during the operation, regardless of the impacts from the surrounding environment. The goal of this thesis has been to solve this problem using gyro stabilization by implementing a stabilization algorithm to be added in the camera unit's software.

The idea behind gyro stabilization is to counteract for external disturbances causing the camera unit to dislocate from its desired attitude. This is realized by controlling two BLDC, Brushless DC, motors in a positioning unit (PU) provided by Axis Communications AB. The PU could be rotated about two axes (using the two motors) perpendicular to each other, pan and tilt, and could hold either a network video camera, or a set of illuminators. The control of the motors is achieved by utilizing data from an IMU, Inertial Measurement Unit, including a gyroscope and an accelerometer, whose measurements are fused in a Kalman filter. This data, together with an estimation of the orientation of the PU, is used to calculate the error between the actual position and the desired position of the PU. The orientation estimation is found by relating the coordinate system of the PU to a fixed, global coordinate system. The calculated error is then minimized by using a PID controller to control the velocity of the pan- and tilt motors and ultimately the position of the PU.

Most of the calculations and simulations regarding the stabilization algorithm were carried out in MATLAB Simulink, and later implemented in the PU's software, written in C code. The final algorithm was tested both on an ABB IRB2400 robot, and mounted on a small boat driven by an outboard motor, made possible by an collaboration with the research program WASP. The results were satisfying and the PU was successfully stabilized, which was the main goal of the thesis. However, there is room for some minor improvements, for example improving the modeling of the motors.

Acknowledgements

Firstly, we would like to express our gratitude to Axis Communications AB and, especially, our supervisor at Axis, Torkel Niklasson for his engagement and assistance during this project. We would also like to say thank you to Mikael Lindberg (Axis) and the team behind WASP, for their work in making our trip to Gränsö possible. Furthermore, a big thanks to Mattias Bognäs (Axis) for helping us design and produce the mount that enabled fastening of the positioning unit onto the robotic arm used during the testing phase of the thesis.

Last but not least, we would like to thank our supervisor at LTH, Professor Anders Robertsson, for his support and valuable feedback during the course of this thesis.

Wilhelm Andrén, Ella Hjertberg
May 2019

List of Abbreviations

API	Application Programming Interface
BLDC	Brushless DC
CPU	Central Processing Unit
DOF	Degree of Freedom
EIS	Electronic Image Stabilization
FPU	Floating Point Unit
HMI	Human Machine Interface
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
LTH	Lunds Tekniska Högskola
MCPU	Main Central Processing Unit
MCU	Microcontroller Unit
MEMS	Microelectromechanical System
MPU-6050	The specific IMU used in this thesis
NED	North-East-Down
PCB	Printed Circuit Board
PID	Proportional Intergral Derivative
PWM	Pulse Width Modulation
UDP	User Datagram Protocol
VAPIX®	Axis' own API Interface
ϕ, θ, ψ	Euler angles roll, pitch and yaw

Contents

1

Introduction

In the past 40 years, it has become possible to, with the emerging camera technology, mount cameras on motorized mounts. This could be useful in several areas, such as search and rescue operations, surveillance or news monitoring. One problem that these applications bring is the difficulty of keeping the camera unit stable during the operation. Parts of this problem could for example be fixed with gyro stabilization, a way of counteracting the disturbances introduced in the mount, in order to stabilize the camera unit platform.

1.1 Background

Axis Communications AB (Axis) is a market-leading company that specializes in network video and audio solutions. Their aim is to innovate *for a safer, smarter world* by combining intelligent technology with human imagination to produce sight-, sound- and analytics-based solutions [Axis, 2019]. Their largest product platform is network video cameras, which range from robust outdoor cameras to smaller, discrete cameras for more sensitive environments. The wide range of applications also mean that there is a wide range of possible mounts, and amongst those also motorized mounts. Some of Axis network video solutions include EIS, Electronic Image Stabilization, which can handle most camera unit disturbances in terms of vibrations. The EIS works efficiently at higher frequencies and low amplitudes, but when the camera unit is affected by low frequency disturbances with higher amplitudes such as undulations, displacements as a result from wind load or wave motions, the EIS cannot compensate. To compensate for these kinds of disturbances, another type of stabilization solution has to be implemented. This implementation could for example be realized using the information from a motion detecting unit, for example a gyroscope and an accelerometer mounted on the camera unit together with a control system to control the motions of this unit. In this way, the low frequency disturbances could be counteracted for and, ultimately, the camera unit stabilized.

A possible solution to this problem has been presented at Axis before, also as a master's thesis [Gustavi and Andersson, 2017], which has been the inspiration for this thesis work, and has laid ground for the theoretical research as well as a foundation for investigating further work. Thus, the motivation for this master's thesis work was to investigate the possibility to develop a stabilization algorithm for an Axis network video unit, more specifically a model named *Axis T99A10 positioning unit 24 V AC/DC* [Axis, 2019], in order to enhance its performance to enable further innovation and progress.

1.2 Goals and problem formulation

The purpose of this thesis is to investigate the possibility of using a positioning unit *Axis T99A10* as a gyro stabilized platform in scenarios where the existing EIS is not sufficient. Examples of scenarios of these types are camera units mounted on ships, tall masts, vehicles, or similar. The aim of the gyro stabilization is to counteract the displacements that occurs when external forces act on the positioning unit, such as disturbances changing the original position of the unit. By counteracting for these disturbances the desired position could be maintained. The desired position could be set by pointing the camera towards an object and fixing that position, to maintain the camera directed towards the object. Another, more general option, is to simply steer the unit so that the camera points in an arbitrary direction, and save this position configuration for the stabilization.

In order to create a gyro stabilization algorithm, an understanding of the different scenarios where it is applicable has to be developed. Therefore, a significant part of the work consists of researching and gathering measurement data from one or two of these scenarios. Another important part in the creation of the stabilization algorithm is choosing the proper equipment necessary for the implementation, which is also included in the research work. Since there are several parts that make up the goal formulation in this thesis, the main goals have been divided into four parts with underlying sub-goals, listed below.

(G1:) Develop an understanding of the different scenarios where gyro stabilization could be used.
Investigation of previously conducted work and thesis work within this subject. This comprises looking into different possible installation scenarios where a gyro stabilization could be of use, to investigate how external forces, such as wind and wave motions, would act on the mounted unit.

(G2:) Choose one or two scenarios based on this research to gather data from and analyze this data for further understanding of the problem.
This part includes deciding which of the scenarios that are interesting for further

investigation, and performing measurements in these chosen scenarios. Another option is to utilize previously gathered data from similar projects, if these are shown to be relevant.

(G3:) Explore available motion detecting units, their program libraries and limitations.
Investigation of the existing motion detecting units included in the positioning unit, as well as alternative options such as other types of sensors than the ones provided. The objective is to find motion detection units suitable for this type of application, and figuring out what requirements to put up in terms of performance for these units. Another sub-goal to this part is to investigate the best placement of the sensors on the positioning unit.

(G4:) Implementation of the stabilization algorithm, using the gathered information and data to develop an algorithm resulting in a prototype of a gyro stabilized platform.
Including an algorithm estimating the unit's orientation and relative coordinate system, performing simulations in a laboratory environment using gathered data and developing a suitable control system to be simulated using MATLAB Simulink. The final prototype is created by implementing the control system in the existing software in the positioning unit, in C code.

1.3 Delimitations

To be able to meet the objectives of this thesis work within the allocated time, some limitations were made in order to set the work load within reasonable limits. Therefore, following assumptions were made during the course of this thesis work.

Assumption 1 The positioning unit together with the camera is in a given position (set by the operator) during the stabilization algorithm. Altering of this position will not be possible during the stabilization algorithm.

Assumption 2 The camera housing in the positioning unit is assumed to be fixed on a frame in an infinite distance from the positioning unit (zoomed-out), to be able to disregard the influence of the disturbances leading to the positioning unit moving in an strictly vertical or horizontal direction.

Assumption 3 The pan- and tilt movements in the implementation can be controlled individually and independent of each other.

Furthermore, focus will not be put on optimizing the implemented code, since the main objective was getting the stabilization algorithm to work properly.

1.4 Course of work

The thesis started off by researching the concept of gyro stabilization which lead to further investigation of different scenarios where gyro stabilization is or could be used. The research was later narrowed down to suitable scenarios for the particular positioning unit. This was followed by investigating different motion detecting unit needed to successfully implement a gyro stabilization algorithm. Some of these units were thoroughly tested before determining which of them were most suitable for the problem in question. The implementation's different parts were performed, designed and tested in parallel. After completing these parts, they were put together into the final stabilization algorithm. The algorithm was then tested and refined during several experiments and finally evaluated according to the main goals of the thesis.

1.5 Individual contributions

To obtain a higher efficiency during the thesis, some of the work was divided between the two participants. Ella focused on designing and testing the Kalman filter, as well as the orientation and reference creation. Wilhelm focused on the modeling of the motors' dynamics and mechanics and the control design. The implementation of the stabilization algorithm in the software was done together, the same goes for the research and investigation of different scenarios and sensors, as well as the report writing.

1.6 Outline

Chapter 1 Gives an introduction to the subject and problem as well as the delimitations that have been made.

Chapter 2 Discloses the materials utilized together with a system overview.

Chapter 3 Presents the theoretical research that this work is based on.

Chapter 4 Presents the research conducted about what sensors to use to succeed with the stabilization algorithm.

Chapter 5 Describes the orientation and coordinate system mapping needed in order to control the positioning unit.

Chapter 6 Describes the modeling and system identification of the dynamical system.

Chapter 7 Presents the research and experiments carried out in terms of designing the control system.

Chapter 8 Features the implementation of the stabilization algorithm, along with code structure for the implementation in the embedded system.

Chapter 9 Contains information about the experiments carried out during the course of the thesis.

Chapter 10 Presents the results.

Chapter 11 Includes discussion and comments on the results as well as conclusions drawn and ideas for further work.

Chapter 12 Presents the conclusions drawn and ideas for further work.

2

Material and system overview

In this chapter, an introduction to the materials used during the course of this thesis work is given, together with a brief overview of the system and its components.

2.1 Platform

The platform that is used for the gyro stabilization implementation is an *Axis positioning unit*, with the model number T99A10 [Axis, 2019], seen in Fig. 2.1. The housing is designed to fit selected Axis network video cameras, and specifically an *Axis PT Illuminator Kit C*, which is a pair of illuminators designed to fit the T99A10. The platform enables jerk-free pan and tilt movements, see Fig. 2.2, and can rotate continuously in pan and from $-90°$ to $45°$ in tilt when the housing is mounted as in Fig. 2.1. The shown position in the figure is called "home" and is defined as (0°, 0°). The possible angle configurations are summarized in 2.1 to 2.3 below. It is possible to mount the interior of the housing (i.e. the network video camera) in the opposite direction, meaning that the tilt range will instead go from $-45°$ to $90°$. The pan movement is constructed with a slip ring, which means that after a full rotation, it will "reset" and start from 0° again.

$$\phi_{pan} = 360° \quad (Limitless) \tag{2.1}$$

$$\theta_{tilt} = \begin{cases} -90° \leq \theta_{tilt} \leq 45° & \text{or} \\ -45° \leq \theta_{tilt} \leq 90° \end{cases} \tag{2.2}$$

$$(\phi_{pan}, \theta_{tilt})_{home} = (0°, 0°) \quad (Homeposition) \tag{2.3}$$

Some of the selected network video cameras that can be mounted inside the housing are equipped with EIS, electronic image stabilization, which stabilizes the image during high frequency vibrations (with low amplitude) of the platform. However, this stabilization is, as previously mentioned, not sufficient when it comes to

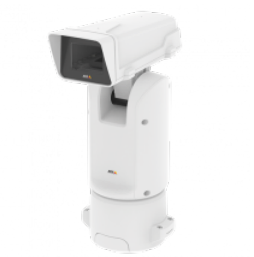


Figure 2.1: The positioning unit *Axis T99A10* [Axis, 2019].

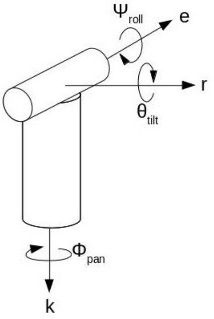


Figure 2.2: Axes of rotation.

oscillations with high amplitude, such as movements from a boat or a car driving in terrain.

Components

The positioning unit T99A10 consists of three main parts: the chassis, the body and the housing. Looking at Fig. 2.3, the chassis is referred to as the lower part of the unit, which is to be fastened onto the mounting environment, and is therefore the part where the disturbances will be introduced in the unit. The body of the unit is referred to as the "middle" part of the unit, which can rotate about its own axis, yielding the pan movement. Furthermore, the housing is the upper part, holding the camera unit or the illuminators, which can rotate about an axis perpendicular to the pan axis, yielding the tilt movement. Two BLDC-motors act as actuators for the pan- and tilt movements, and are both installed in the body of the unit. The tilt motor is connected with the camera housing using a belt gear, and the pan motor is installed using a slip ring, to allow for limitless pan rotation. Each motor is also connected to a motion controller, which handles the trajectory generation of the movements, in acceleration, position and velocity. The motion controller's role is to induce a smoother movement. The speed- and acceleration limits of these motors are shown

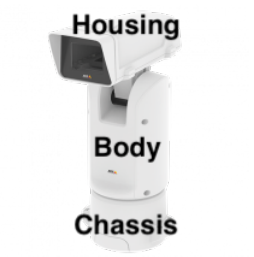


Figure 2.3: Positioning unit *Axis T99A10*, with named parts (adapted from [Axis, 2019]).

below in 2.4 to 2.6, and are set equally for both motors. These limits are the default limits of the unit, which were shown to fit the purpose of this thesis.

$$-200°/s^2 \leqslant a_{pan,tilt,max} \leqslant 200°/s^2 \tag{2.4}$$

$$-120°/s \leqslant \frac{da_{pan,tilt,max}}{dt} \leqslant 120°/s \tag{2.5}$$

$$-0.05°/s \leqslant \frac{da_{pan,tilt,min}}{dt} \leqslant 0.05°/s \tag{2.6}$$

Furthermore, there are two PCB:s, Printed Circuit Boards, located in the body and in the chassis of the unit, respectively, containing the electric components. Mounted on the PCB:s are the available sensors, and the MCU, Microcontroller Unit, which holds the software of the positioning unit, and is also where the stabilization algorithm should be implemented. Furthermore, the different components in the PCB handles the communication with and control of the motors and the motion controllers, amongst others.

EIS

The camera unit installed in T99A10 includes an Electronic Image Stabilization, which cancels out jitter in the image frame, such as high frequency vibrations, for example hand or vehicle jitter. The built-in gyroscopic sensors in the camera unit detect the vibrations and motions, and by cutting of pixels in the image frame and automatically adjusting it, the smooth frame can be created [Axis, 2019].

Important to note here is that there exists a gyroscopic sensor in the network camera, which is not the same as the gyroscope that is used to implement the stabilization algorithm in this thesis.

To learn how much the EIS could compensate for in terms of vibrations, some trigonometric calculations were carried out, showing how large vibrations can occur without the image frame getting distorted. When the camera frame is maximum zoomed, the field of view is 2°[Axis, 2019], and at this point the EIS can handle a distortion of approx. 3m, which corresponds to a displacement of 0.3086 mm at the lens. In terms of angular changes, 0.3086mm could be converted into approx. 0.104°, which clearly indicates the need for another stabilization algorithm, that could handle greater disturbances.

Sensors

There are different types of sensors available for motion detection in order to estimate the positioning unit's orientation. The sections below describe the different types of sensors that have been investigated during this thesis work, which are:

- IMU, Inertial Measurement Unit, consisting of a three-axis gyroscope and a three-axis accelerometer
- IMU consisting of the sensors above, with an additional three-axis magnetometer.
- Pan- and tilt encoders

In the T99A10, there are two PCB footprints for an IMU named MPU-6050 [TDK InvenSense, 2019a]. A footprint means that a component could be installed without having to do any alterations to the existing PCB. The footprints in T99A10 are placed in the base and body PCB:s of the unit, respectively.

Gyroscope A Gyroscope is a device used to measure rotational motion. One type of gyros are MEMS (Microelectromechanical System) gyros. Gyros of this sort are small sensors that measure angular velocity in the unit degrees per second ($°/s$) and revolutions per second (RPS). A triple axis MEMS gyroscope, which can be found in the MPU-6050 described above, can measure the angular velocity and rotation around three different axes; x, y and z [SparkFun Electronics, 2013b]. Fig. 2.4 shows the different axes and the rotation around these axes of a triple axis MEMS gyroscope (in this case the MPU-6050).

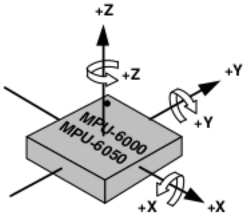


Figure 2.4: Orientation of axes of sensitivity and polarity of rotation of the MPU-6050 [TDK InvenSense, 2019a].

When selecting a specific gyroscope, the measurement range of the sensor needs to be taken into consideration. The measurement range, or full-scale range, is the maximum value of angular velocity of which the gyro can measure. Depending on what is measured, the chosen range can vary [SparkFun Electronics, 2013b]. The MPU-6050 has a full-scale range of ± 250, ± 500, ± 1000 and $\pm 2000°/s$ [TDK InvenSense, 2019a]. A drawback with using gyroscopes for measurements are that they are prone to drift. This means that the initial zero reading of the gyroscope will drift over time due to integration of internal imperfections and noise within the device [Beavers, 2017]. There is also a constant bias offset in the axes of the gyroscope, which has to be measured and compensated for prior to use.

Accelerometer An accelerometer is a device used to measure acceleration, i.e. the time rate of the velocity of a specific object. The measurements are in the unit meters per second squared (m/s^2) or in G forces (g). On Earth, a single G-force unit (1g) is equivalent to approximately $9.82 m/s^2$ but can vary slightly due to elevation and different gravitational pull. The accelerometer measures the acceleration along three axes, see Fig. 2.4. The acceleration measured can either be static or dynamic forces. Static forces of acceleration include gravity while dynamic forces can include vibrations and movement. Accelerometers also have an eligible range of forces of which they can measure. Generally, a smaller range results in a higher sensitivity of the measurements [SparkFun Electronics, 2013a]. The MPU-6050 has a range of ± 2, ± 4, ± 8 and ± 16g [TDK InvenSense, 2019a].

A drawback with accelerometers is that they are sensitive to vibrations and mechanical noise [SparkFun Electronics, 2013a] leading to noisy measurements that are not fully correct or intuitive to interpret.

Magnetometer A magnetometer is used to measure magnetism. The measurements can include magnetization of a magnetic material or the strength, direction or relative change of the magnetic field at a specific location. The magnetometer can therefore be used to measure the direction in which the sensor (and further on the unit on which the sensor is installed) is oriented, in similar with a compass [ScienceDirect, 2019].

One IMU including a magnetometer is the MPU-9250 which contains a 3-axis magnetometer and can therefore measure the magnetism in three axes; x, y and z. The orientation of these can be seen in Fig. 2.5. The magnetism is measured in the unit micro Tesla (μT). The magnetometer in the MPU-9250 has, just like the gyroscope and the accelerometer, a full-scale measurement range which is preset to $\pm 4800 \mu T$ [TDK InvenSense, 2019b].

A magnetometer's bias is caused by distortions which are divided into two types: hard and soft iron distortions. Hard iron distortions are caused by objects, placed near the sensor, that create a magnetic field. Soft iron distortions are considered deflections or alterations of the magnetic field. One way to determine the bias and these distortions is to read and plot the output of the magnetometer while slowly rotating the magnetometer around its z axis. An example of a plot of this type is

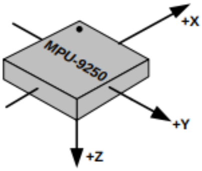
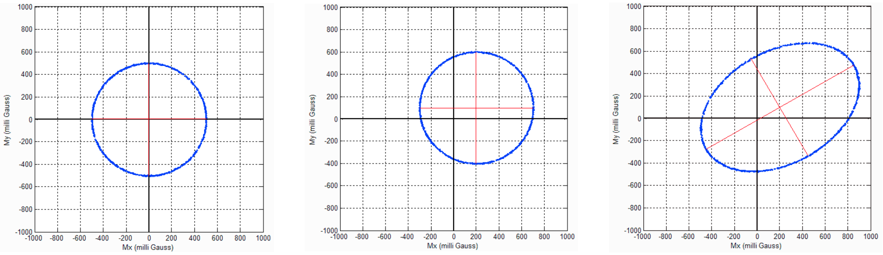


Figure 2.5: Orientation of axes of sensitivity of magnetometer [TDK InvenSense, 2019b].

seen in Fig. 2.6, showing a general case of a magnetometer with different distortions [Vectornav, n.d.]. If there are no distortions present, the plot should show a circle centered in x = 0 and y = 0, as can be seen in Fig. 2.6a. Fig. 2.6b shows how hard iron distortions causes the center of the plotted circle to move. Lastly, as seen in Fig. 2.6c, when there are both hard and soft iron distortions, the circle center is moved and the circle itself is deformed [Vectornav, n.d.].



(a) Output with no distortions.

(b) Output with hard iron distortions.

(c) Output with hard and soft iron distortions.

Figure 2.6: Plotted magnetometer output with different cases of distortions [Vectornav, n.d.].

It is important that the bias of the sensors are taken into account and corrected when reading from the sensors. Therefore, the sensors must be calibrated. One way to calibrate the sensors is to measure the bias during an initiation phase and then erasing it [SparkFun Electronics, 2013b].

Encoders Two encoders are placed in the positioning unit, one for each motor. The encoders convert information from the motors into a corresponding position in degrees, relative the motor's original position [Tala, 2014]. The encoders utilizes hall effect sensors to present the correct information, which are relevant when the current position of the pan- and tilt-motors is desired.

2.2 System architecture

The system architecture of the positioning unit consists of the motor control, IMU and the operator control and is graphically described in Fig. 2.7. The operator sends commands to the positioning unit from an HMI, e.g. from a joystick connected to a PC. The HMI communicates with the unit via VAPIX®, which is Axis' own API [Axis, 2019], and sends the commands to the main CPU, on the PCB, which runs the embedded operating system. The main CPU communicates the signals to the MCU, which is the microcontroller unit, via a serial port, which handles the data collection from the IMU. The system regarding the EIS is not included in this scheme, since it is an individual system incorporated in the network camera. Furthermore, the MCU forwards the operator commands via a PWM to the BLDC motor drivers, which controls the pan- and tilt motors. From the hall effect sensors incorporated in the BLDC motordrivers, encoder data is received and sent to the motion controller inside the MCU. The BLDC motors are connected to a gear box and belt gear managing the gear ratio of the motors. This system is designed to always represent the correct encoder data, even if the motors were to fail.

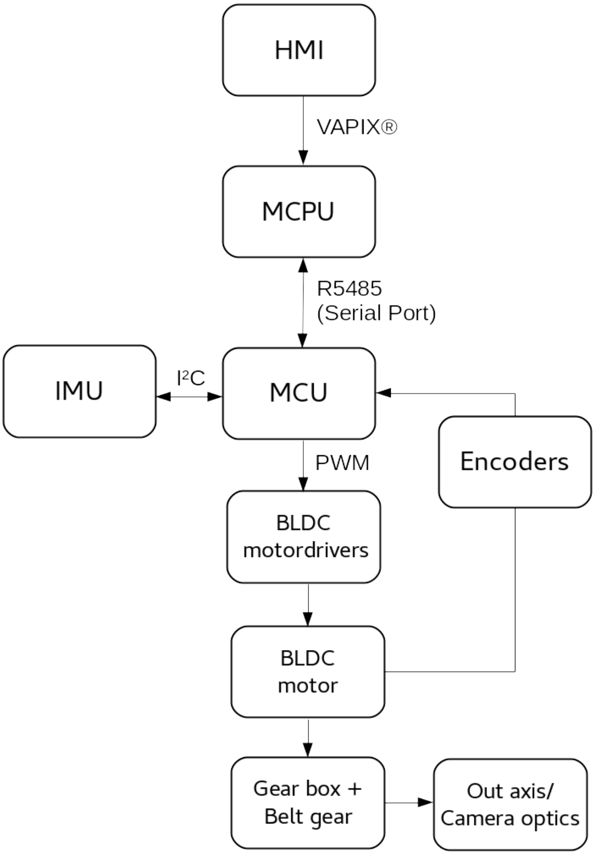


Figure 2.7: System architecture of T99A10.

3

Scenarios

In order to succeed with the gyro stabilization algorithm, the different parts that contribute to the implementation must be investigated and understood. The following section describes the research carried out in terms of investigating relevant scenarios where gyro stabilization could be used. Scenarios where stability disturbances could occur due to external forces or other influences were the ones considered relevant. After consulting supervisors and due to the limited amount of time, the research was limited to three main scenarios: *marine environments*, *terrain* and *wind load*. The scenarios were investigated for the suitability and possible application of gyro stabilization. All three scenarios seemed equally interesting in terms of gyro stabilization, since the effect from all these scenarios is making it difficult to maintain a stable platform and therefore requires some sort of stabilization. However, relevant measurement data was only successfully collected in two of the scenarios, marine environments and terrain, which are presented below. Since gathering relevant measurement data was important in order to further develop the stabilization algorithm, the wind load scenario was disregarded.

3.1 Marine environments

In marine environments, aqueous vehicles like ships and boats are affected by different disturbances due to waves. There are a few types of waves depending on the relative angle between the wave itself and the ship. These types are following and head waves that encounter the ship at an angle of 0° and 180°, respectively, quartering and bow waves at 45° and 135°, respectively and beam waves at 90°. Fig. 3.1 shows the definition of the relative wave angles [Perez, 2005]. As a ship encounters a wave of any type, a motion in any of the ship's degrees of freedom (DOF) will occur. A ship has got 6 DOF: surge, sway and heave represent motions along the x-, y- and z axis, respectively. Roll, pitch and yaw represent rotations around the surge-, sway- and heave axis, respectively [Perez, 2005]. Fig. 3.2 describes these different DOF. Various factors affect the ship's motions when encountering a specific wave. Characteristics of the ship such as weight, size and stiffness of the keel are included

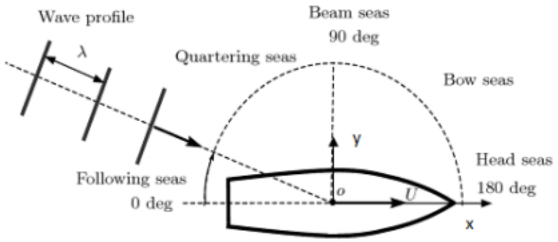


Figure 3.1: Different waves due to different relative angle between wave and ship [Perez, 2005].

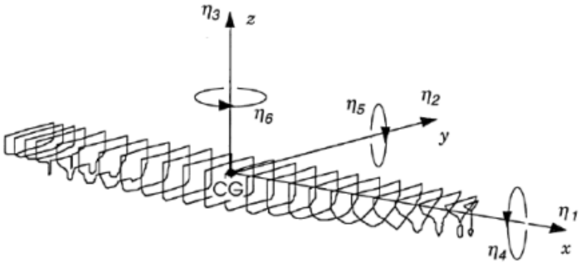


Figure 3.2: A ship's degrees of freedom (DOF) [Hua and Palmquist, 1995].

in these factors as well as the wave's particulars such as amplitude, frequency and relative angle to the ship [Ovegård, 2009]. Depending on these factors, the ship's motions will vary over the different DOF. When studying the motions, one can determine that the ship, in some cases and DOF, follows the wave's motions along the water. For example, a ship motion in the heave direction usually follows the wave's motions and amplitude while a rolling motion of the ship follows a different motion rather than the wave motions [Bergdahl, 2009].

Due to the different types of waves and encountering angle of the waves, disturbances occur which causes the ship to behave differently than just following the motions of the wave. These disturbances are usually represented as low frequency vibrations on the whole ship, in different directions and rotations, at approximately 0.01-1.5 Hz depending on the conditions of the water.

Some experiments investigating these phenomenons have been carried out at Axis, and in Figs. 3.3 and 3.4 these experiments are shown. The plots show the measurements from an IMU placed on a boat, and recording while the boat was driving at sea. From these figures, it is seen that the boat is subject to displacements larger than the EIS could handle, which was approx. $0.1°$. It could also be seen that the angular speed caused by the disturbances (wave motions) does not exceed the speed limits, which was 120°/s in the pan- and tilt motors, though it comes quite close at certain points.

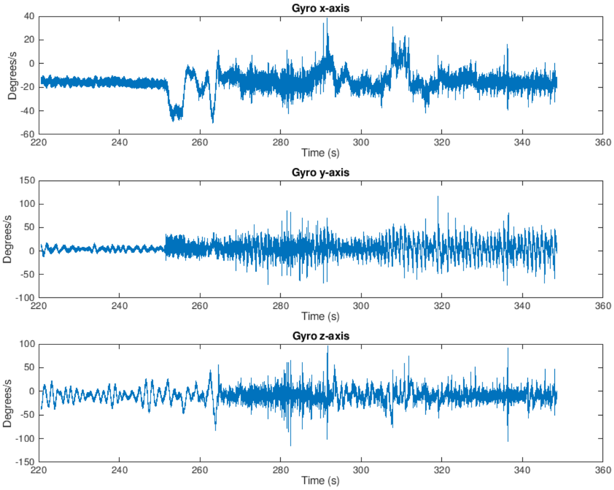


Figure 3.3: Example of angular velocities measured by the gyroscope on a boat at sea.

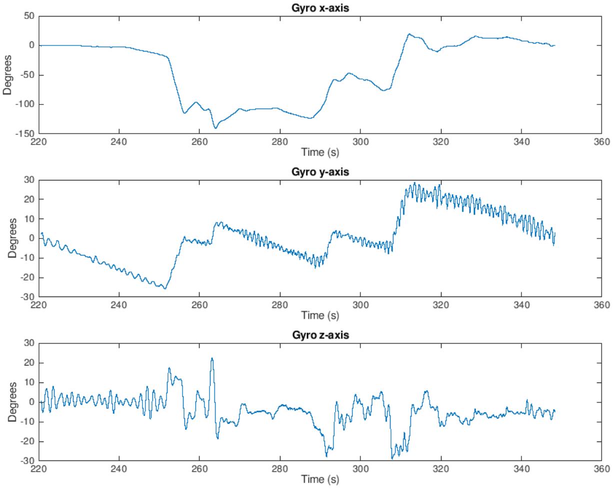


Figure 3.4: Example of angular displacements from integrated gyroscope measurements on a boat at sea.

3.2 Terrain

In order to determine how a vehicle will behave in terrain environments, the interaction between the vehicle and terrain has to be investigated. The study of this interaction is often referred to as *Terramechanics* [Drescher and Hambleton, 2010]. The characteristics of the surface material play an important role in determining the vehicle's performance, trafficability and soil deformation of the vehicle passage [ISTVS, 2019]. Terrain is referred to as the soil, vegetation and topography of the ground, and trafficability means the ability of the terrain to support a vehicle in terms of providing traction [ISTVS, 2019]. There is no general way to determine these variables, since there are many other factors contributing to the result, such as the damping of the wheels, the specific construction of the vehicle's body, etc. However, there is a possibility to use predictive models, where rough assumptions about the terrain and topography should be made, but this methodology is not further analyzed in this report. Instead, velocity- and position data collected from gyroscope measurements in terrain was investigated, where Fig. 3.5 shows the orientation of the rotation axes in a car driving in terrain. In Figs. 3.6 and 3.7 the data from the aforementioned experiment is shown. These experiments were performed at Axis [Axis, 2019] by placing an IMU inside a car and collecting measurements while driving the car around in different terrain and with varying speed. From the figures it can be seen that the amplitude of the disturbances exceeds the stabilization ability of the EIS. Regarding the angular velocities, they are within the limits of the pan- and tilt motors, and could thus, theoretically speaking, be counteracted by the motors.

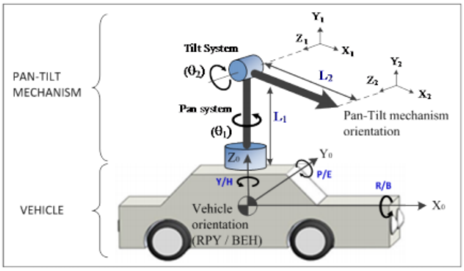


Figure 3.5: Relation of coordinate systems [Saputra et al., 2013].

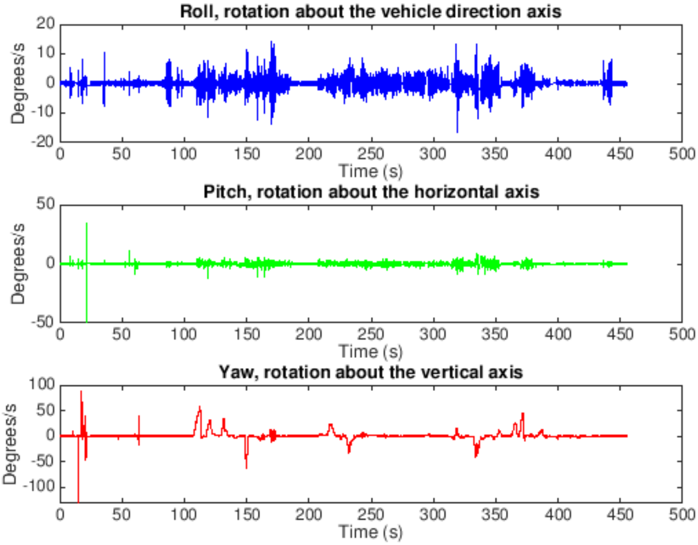


Figure 3.6: Example of angular velocities measured by the gyroscope on a car driving in terrain.

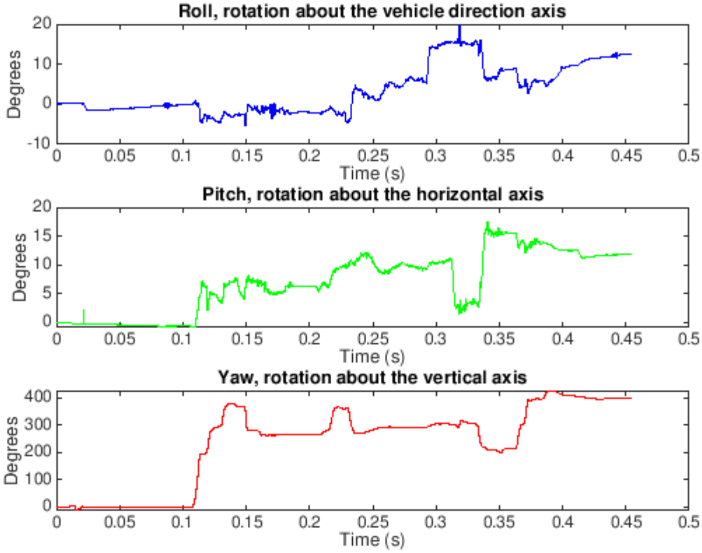


Figure 3.7: Example of integrated gyroscope measurements (angular displacements) from a car driving in terrain.

4

Choice and placement of sensors

In order to determine which sensors that were most suitable for the application, and the placement of these sensors, several tests along with extensive research were carried out. Three different types of sensors, mentioned above in section 2.1, was investigated for the choice; a gyroscope, an accelerometer and a magnetometer. These sensors are included in the two, previously described IMU:s: the *MPU-6050* and the *MPU-9250*.

4.1 Validating the MPU-9250

The magnetometer in the MPU-9250 measures magnetism in μT in the x-, y- and z direction. When first measuring, the bias must be compensated for. Therefore, the read values in the three directions were scaled according to 4.1,

$$\begin{aligned} X_{Scaled} &= \frac{X_{Raw} - X_{Mean}}{X_{Max} - X_{Mean}} \\ Y_{Scaled} &= \frac{Y_{Raw} - Y_{Mean}}{Y_{Max} - Y_{Mean}} \\ Z_{Scaled} &= \frac{Z_{Raw} - Z_{Mean}}{Z_{Max} - Z_{Mean}} \end{aligned} \tag{4.1}$$

where X_{Scaled}, Y_{Scaled} and Z_{Scaled} are the scaled magnetometer values in the three directions, X_{Raw}, Y_{Raw} and Z_{Raw} the raw magnetometer values in the three directions, X_{Mean}, Y_{Mean} and Z_{Mean} the mean values in the three directions and X_{Max}, Y_{Max} and Z_{Max} the maximum values in the three directions. The mean value is calculated according to 4.2,

$$Mean = \frac{Max + Min}{2} \tag{4.2}$$

where Max is the maximum value and Min the minimum value. When scaling, the resulting values should be inside the interval $[-1:1]$ in order to more easily study the data. After scaling the read values from the magnetometer, the relative angle between two axes must be measured in order to determine the heading/direction the magnetometer is oriented in. This should be done since the magnetometer could be used as a compass to determine the yaw angle of the positioning unit. Depending on which direction the magnetometer is installed, the relative angle will be measured between different axes. The relative angle is measured between, for example, the x- and y axis according to 4.3,

$$\psi = atan2(X_{Scaled}/Y_{Scaled}) * (180/\pi) \quad (^\circ) \tag{4.3}$$

where ψ is the heading angle of the sensor. $atan2$ denotes the angle in the Euclidean plane, in radians, between the positive x axis and the ray to the point on the unit circle $(x,y) \neq (0,0)$ [Mathworks, 2019]. The possibility of using a magnetometer combined with a gyroscope and an accelerometer was investigated by studying the measured values of the MPU-9250. The main investigation was to determine whether the magnetometer was affected by the magnetic fields caused by the motors and the heater of the positioning unit. Therefore, different scenarios for testing the magnetometer were established. These scenarios included different placements for the sensor on the unit as well as running the different motors and toggling the heater. The chosen placements were on the PCB at the body of the unit, located near the MCU and the tilt motor, and on the PCB in the chassis of the unit. The two placements were chosen to be located as near the existing footprints for the IMU:s as possible. The different motor scenarios were *stationary*, *running tilt motor*, *running pan motor* and *running pan and tilt motor*. When toggling the heater, the motors were kept stationary. These experiments are presented in the two following segments.

Experiment 1: Magnetometer placed in the body of the unit

The magnetometer was not affected by the power supply when the positioning unit was turned on and kept stationary, which is why these results are not shown. However, it was affected during the tilt movement which can be seen as the oscillating curves on all axes in Fig. 4.1 which was rather expected since the placement was adjacent to the tilt motor, thus close to the tilt motor's magnetic field. The magnetometer was affected by the pan movement, seen in Fig. 4.2, which was also quite expected since the magnetometer itself rotates together with the pan movement. The way the magnetometer was placed resulted in that a pan movement would yield a rotation about the magnetometer's y axis, which corresponds with the results. During a movement in both pan and tilt, the magnetometer behaved the same, no disturbances from the pan movements but from the tilt movements. When the heater was toggled and the unit kept stationary, the magnetometer was affected by an offset in the x- and z axis, which can be explained by the fact that another magnetic field is introduced when the heater is turned on, causing a hard iron distortion to occur on the magnetometer.

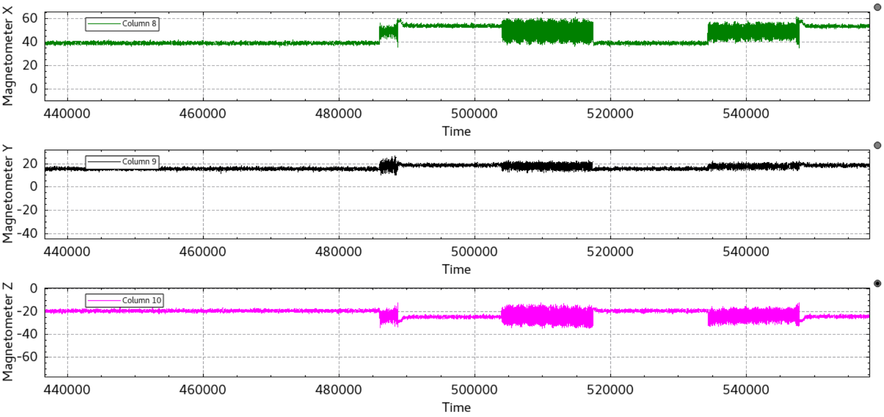


Figure 4.1: Placement 1: Magnetometer measurements for the positioning unit during tilt movement.

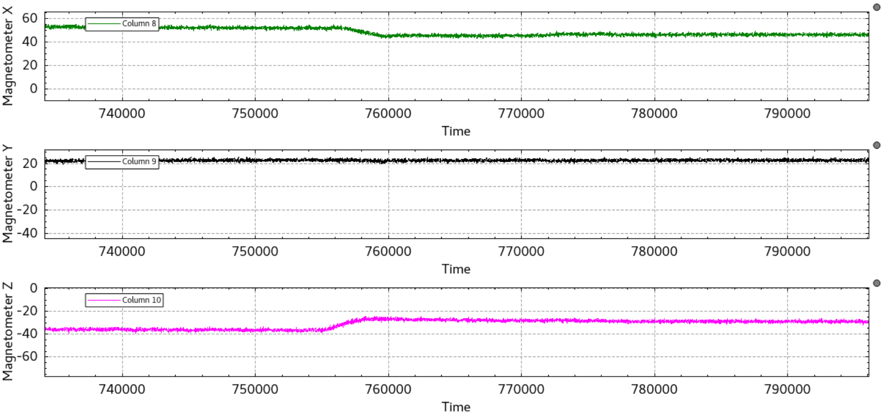


Figure 4.2: Placement 1: Magnetometer measurements for the positioning unit during pan movement.

After studying the results, it was concluded that the offset caused by the heater was not a problem for using the magnetometer since it could be compensated for during calibration. However, the impact from the tilt movement would cause problems since the measurements were disturbed by oscillations. This would in turn cause misleading measurements and estimations of the compass heading. The final conclusion was that the MPU-9250 could not be installed at placement 1.

Experiment 2: Magnetometer placed in the chassis of the unit

The same five use cases as for placement 1 was conducted for the second placement alternative, where the MPU-9250 was placed on the PCB in the chassis of the positioning unit. The magnetometer was not affected by the power supply when the positioning unit was turned on, same as in placement 1, neither was it affected during tilt movements. This was expected since the tilt motor is located far away from the magnetometer in this particular placement, i.e. its magnetic field does not disturb the measurements.

When the pan motor was running, some distortions in the magnetometer measurements was noted in all axes, as can be seen in Fig. 4.3. This concluded that the magnetometer was affected by the magnetic field caused by the pan motor. The same behaviour occurred when both pan and tilt motors were running. With the unit in idle, the heater was toggled which resulted in the magnetometer measurements in Fig. 4.4.

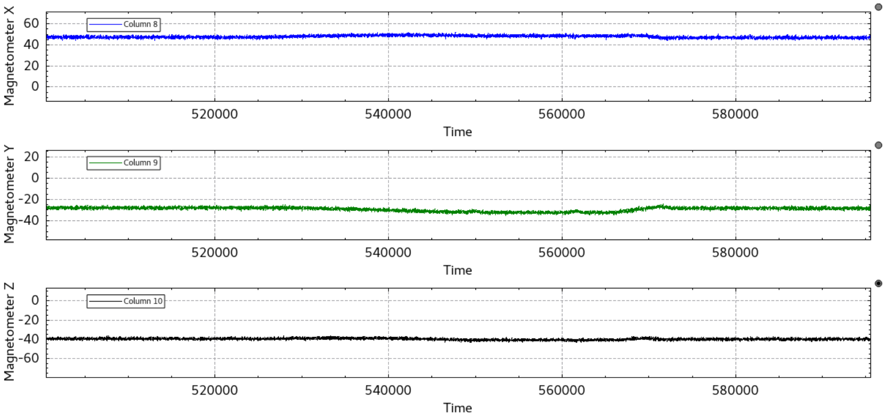


Figure 4.3: Placement 2: Magnetometer measurements for the positioning unit during pan movement.

Turning the heater on leads to a greater load on the power supply, which is located in the base of the unit. Therefore a hard iron distortion occurs on the magnetometer, causing an offset in the measurements. This is however manageable since the offset could be eliminated in the implementation. What cannot be managed is the distortion that is created by the magnetic field in the pan motor. Using the magnetometer as a compass will therefore lead to misleading and undesired results, why the conclusion was made that the magnetometer could not be placed in the chassis of the unit, and neither used in this configuration at all.

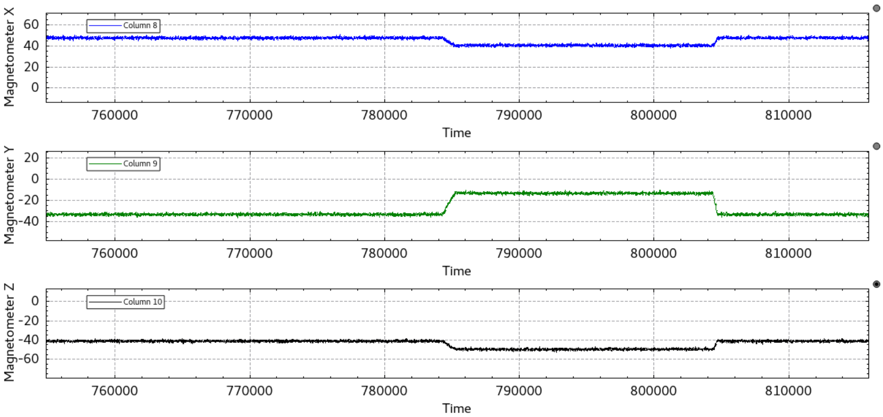


Figure 4.4: Placement 2: Magnetometer measurements for the positioning unit the heater toggled.

4.2 Investigating the MPU-6050

After deciding to disregard the MPU-9250, the other IMU with a footprint in the PCB, MPU-6050, was further investigated. These investigations were made in order to determine whether the MPU-6050 was the best possible IMU for solving the problem. Also, the placement of this IMU was investigated in order to achieve the best results possible. The placements taken into consideration were the same as for the MPU-9250, i.e. at the body and in the chassis of the positioning unit.

Performance

When choosing an IMU, some specifications need to be taken into consideration. These include the measurement range, the number of axes and the interface of the gyroscope and accelerometer. The range of the gyroscope and accelerometer, as mentioned earlier, is the upper and lower limit of which the sensors can measure. The goal is to match the range of the sensor so that the actual measurement values, lies well within this range. Since the MPU-6050 features a range for both the gyroscope and the accelerometer which can be chosen to a wide set of values, the IMU is believed to be able to prove the concept of the thesis, in a range perspective [TDK InvenSense, 2019a].

The number of axes of the gyroscope and the accelerometer means how many, out of the three axes, x, y and z, the sensors can measure [SparkFun Electronics, n.d.]. The MPU-6050 includes both a gyroscope and an accelerometer that can measure in all three axes and is therefore, considered a good choice of sensor [TDK InvenSense, 2019a].

The interface of the IMU means the method by which data is sent and received between a controller and a device. The interface is often divided into analog and digital interface, both with its own advantages and disadvantages [SparkFun Electronics, n.d.]. The MPU-6050 features an I^2C digital interface [TDK InvenSense, 2019a], which is a common interface in situations where multiple axes need to be read out to a controller. Accelerometers with an I^2C interface are also less susceptible to noise than analog accelerometers [SparkFun Electronics, n.d.]. The MPU-6050 is therefore considered a good choice from this perspective as well.

Placement of the MPU-6050

The placement of an IMU depends on various factors that will affect the IMU's accuracy and performance. One of these factors is the thermal impact on the IMU due to placements nearby heat sources such as circuits from power management, processors or other high-current carrying devices. InvenSense MPU devices, like the MPU-6050, features temperature compensation, but variations in device temperature should be avoided. This is due to the fact that variations can cause changes in sensor accuracy [InvenSense, 2014]. After reasoning with the supervisor at Axis, the conclusion was made that the placement with the least temperature fluctuations was at the body of the positioning unit.

Another factor to consider was to place the IMU as near to the center of gravity as possible. This is to minimize the need for position offset compensation for the measurements. If such compensation is required, the overall accuracy of the IMU is reduced. The center of gravity is the point of rotation of the entire structure [Basic Air Data, 2011]. This means that if the positioning unit were to be installed on e.g. a ship, the center of gravity would end up at the rotation center of the ship itself. Therefore, it does not matter whether the placement of the IMU is at the body or the chassis of the positioning unit since the difference in distance to the center of gravity is negligible.

A factor that was more specific to this thesis was the eventual latency that occurs when placing the IMU at the chassis of the positioning unit. When the IMU is placed at the chassis of the unit, it has to communicate to a specific MCU located at the chassis as well. The communication between the main MCU in the body and the IMU will therefore require an extra step of communication, causing a slight decrease in communication speed.

When the aforementioned factors was taken into consideration, the final conclusion was to place the IMU in the footprint at the body of the positioning unit.

4.3 Investigating the camera-mounted IMU

Another alternative that arose during the investigation was the possibility of using the IMU that was internally placed inside the camera in the housing of the positioning unit. This IMU was the same as the one placed in the footprint of the positioning unit, an MPU-6050. The initial idea was that this placing would have facilitated the calculations since the number of coordinate systems would have decreased. While further discussed, it was realized that this placement would lead to an additional step in the communication with the main MCU, which could imply latency in the implementation. Since this quantity is difficult to measure, the robustness of this placement was investigated instead. Since the housing and the camera is at the extremity of the positioning unit, it seemed reasonable that this part was more sensitive to vibrations and disturbances than the body of the positioning unit. Therefore, it was argued that the IMU measurements would be of higher quality in the body of the unit, to be able to in greater extent decrease the measurement noise. D, the camera-mounted placement alternative was neglected without further investigation.

5

Coordinate systems and orientation

In order to stabilize the positioning unit and counteract for external influences, the current orientation of the positioning unit in relation to its desired orientation has to be determined. The camera housing is the part of the positioning unit that should ultimately maintain a desired attitude, for example pointing towards an object. In order to describe this attitude, the coordinate system of the housing could be related to a global coordinate system that is fixed in the earth frame. For this purpose, the North-East-Down (NED) frame was chosen, with "Down" aligned with the direction of the gravity vector, see Fig. 5.2 [CHRobotics, n.d.]. The chassis of the positioning unit is fixed and mounted on the moving object, which means that the chassis orientation cannot always be measured in the NED frame. The same reasoning applies for the IMU, which measurements are relative to its mounted position. As a result from this, there will be three separate coordinate systems that should be related to one another in order to describe the positioning unit's orientation at all times. These are presented in the list below and visualized in Fig. 5.1.

- NED - the global coordinate system, fixed in the earth frame.
- IMU - the IMU's coordinate system.
- chassis - the coordinate system of the chassis, where the disturbances enter.

The following section will present a suggestion of how to reach this orientation description.

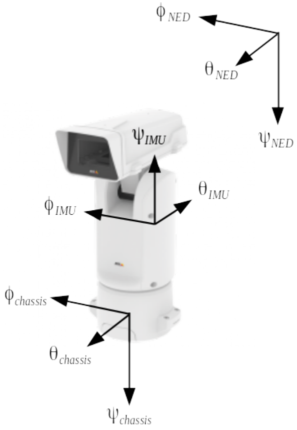


Figure 5.1: Coordinate system representation in the positioning unit.

5.1 Euler angles

To represent an object in a 3D orientation with respect to a fixed frame, Euler angles can be used. Euler angles use a combination of three rotation axes, roll (ϕ), pitch (θ) and yaw (ψ), see Fig. 5.2, to describe the 3D orientation by basic rotations about these axes [Freidovich, 2017]. Each rotation can be described by a rotation matrix, 5.1 to 5.3, where each matrix represents the mapping from one frame to another. The resulting rotation matrix 5.4 shows the mapping from the inertial frame (which is the frame from which the rotation starts) to the frame of the chosen object relative to that frame, for example the IMU frame or the chassis. Below, the denotations 1 and 2 represent the "mid-result" rotations about the separate axes, while O denotes the object to which the inertial frame is related. The mapping starts with a yaw rotation in the inertial frame, shown in 5.1, resulting in frame 1, $R_{NED}^{1}(\psi)$. The next step is a pitch rotation in frame 1, 5.2, which yields frame 2, $R_{1}^{2}(\theta)$. Finally, a roll rotation in frame 2 is made, 5.3, reaching the object frame O, $R_{2}^{O}(\phi)$ which is aligned with the coordinate system of the chosen object. To go from the inertial frame to the object frame, the rotation matrices are multiplied, which results in 5.4.

$$R_{NED}^{1}(\psi) = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1}$$

$$R_{1}^{2}(\theta) = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{5.2}$$

$$R_2^O(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{5.3}$$

$$\begin{gathered} R_{NED}^O(\phi,\theta,\psi) = R_2^O(\phi)R_1^2(\theta)R_{NED}^1(\psi) \\ = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ c\psi s\theta s\phi - c\theta s\psi & c\theta c\psi + s\theta s\phi s\psi & c\theta s\phi \\ c\phi c\psi s\theta + s\phi s\psi & c\phi s\theta s\psi - c\psi s\phi & c\theta c\phi \end{bmatrix} \end{gathered} \tag{5.4}$$

Using this method enables relating separate coordinate frames to each other in an intuitive way [CHRobotics, n.d.]. Note that in the NED frame in Fig. 5.2, *pan* and *tilt* coincides with *yaw* and *pitch*, respectively. For the sake of simplicity, the latter convention will be used to describe rotations further on. A risk with using Euler angles is the occurrence of a phenomenon called *Gimbal lock*, which is when an orientation cannot be uniquely described using Euler angles. Depending on the order in which the rotations are computed, this occurs at different orientations, which with the order presented above is when the pitch angle is 90° [CHRobotics, n.d.]. In the pan and tilt-notation, this means that the housing cannot be represented by Euler angles if the housing is tilted 90° downwards, in its utmost position. To avoid gimbal lock, an alternative method to Euler angles can be used, called quaternions. Even though gimbal lock could be a risk for this application, though it is non-likely, the use of quaternions was discarded in this thesis work.

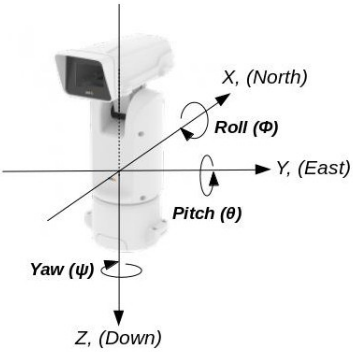


Figure 5.2: positioning unit orientation in the NED frame (Figure adapted from [Axis, 2019]).

5.2 Extracting Euler angles from rotation matrices

When the rotation matrices are derived, the Euler angles could be extracted from the rotation matrices to get the orientation representation in *roll*, (ϕ), *pitch*, (θ), and *yaw*, (ψ). Considering a standard rotation matrix on the form 5.5 (same as 5.4), the corresponding Euler angles can be extracted with 5.6 to 5.8 [Day, n.d.],

$$R = \begin{bmatrix} R_{01} & R_{02} & R_{03} \\ R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \end{bmatrix} \tag{5.5}$$

$$\phi = atan2(R_{13}, R_{23}) \tag{5.6}$$

$$\theta = atan2(-R_{03}, \sqrt{R_{01}^2 + R_{02}^2}) \tag{5.7}$$

$$\psi = atan2(sin(\phi)R_{21} - cos(\phi)R_{11}, cos(\phi)R_{12} - sin(\phi)R_{22}). \tag{5.8}$$

The sign of the parameters in each extraction may vary with the appearance of the current coordinate system. For example, if the yaw axis in the inertial frame is positive in the downwards direction and the yaw axis in the object frame is positive in the upwards direction, the sign in 5.8 has to be changed in order for the yaw angle to be represented in the correct way in the object frame.

Euler angle extraction from IMU readings

In order to get the IMU measurements represented in the global NED frame, they have to be converted from the frame in which they are read. This is performed using the matrix calculations shown above in 5.1 to 5.4. To calculate the accelerations in the NED frame from accelerometer readings made in the object frame, the acceleration vector from the object frame is simply multiplied with the rotation matrix from the object frame to the NED frame, as in 5.9 [CHRobotics, n.d.], and the Euler angles can then be extracted using the equations mentioned in the previous subsection.

$$v_{NED} = R_O^{NED}(\phi, \theta, \psi)v_O \tag{5.9}$$

where $v_{NED} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$. Notice here that the yaw angle cannot be extracted from the accelerometer readings, since a rotation about the gravitational axis will not yield any effect on the accelerometer. The same approach as for the accelerometer measurements does not apply for gyroscope measurements. Since the measurements from the gyroscope are angular rates, they cannot be converted to the NED frame using the same rotation matrix. Instead, the following rotation matrix must be used [CHRobotics, n.d.],

$$D(\phi,\theta,\psi) = \begin{bmatrix} 1 & sin\phi tan\theta & cos\phi tan\theta \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi/cos\theta & cos\phi/cos\theta \end{bmatrix}$$

which converts the gyroscope readings to each of the yaw, pitch and roll axes. The z axis gyroscope reading should be rotated to the inertial frame, the y axis gyroscope reading should be rotated to the frame denoted 1 and the x axis gyroscope reading should be rotated to the frame denoted 2 [CHRobotics, n.d.]. The gyro rates converted into the NED frame are then retrieved from 5.10,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = D(\phi,\theta,\psi) \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} p + qsin\phi tan\theta + rcos\phi tan\theta \\ qcos\phi - rsin\phi \\ qsin\phi/cos\theta + rcos\phi/cos\theta \end{bmatrix} \tag{5.10}$$

where *p* represents the x axis gyroscope reading in the object frame, *q* represents the y axis gyroscope readings in the object frame, and *r* represents the z axis gyroscope readings in the object frame. To derive the corresponding Euler angles from these rates, the converted gyro data has to be integrated.

6

Modeling

This section includes a more profound presentation of the motors in the positioning unit. Based on this theory, modeling of the motors was performed, which ultimately laid ground for the control system of the stabilization algorithm.

6.1 Brushless DC motors

The motors in the positioning unit T99A10 are brushless DC electric motors (BLDC motors). These motors are permanent magnet synchronous motors that use an inverter and position detectors to control the armature currents. A BLDC motor's operating characteristics resemble those of a conventional DC motor but instead of using a mechanical commutator, it employs an electronic commutation which makes it a virtually maintenance-free motor [Sen, 1997]. There are several different configurations of a BLDC motor but the most common one, and the one used in the positioning unit, is the three-phase motor due to its high efficiency and low torque ripple [Texas Instruments Inc., 1997]. Fig. 6.1 shows the transverse section of a BLDC motor. The position detection is mainly implemented using three Hall effect sensors that detect the presence of small magnets located on the motor shaft.

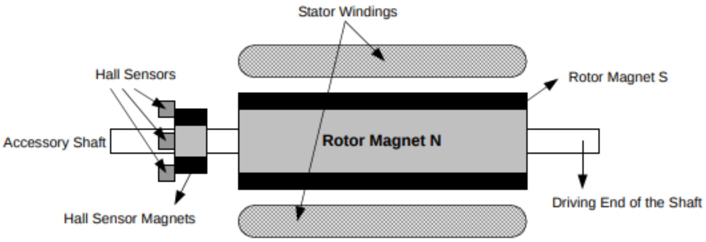


Figure 6.1: Transverse section of a BLDC motor [Yedamale, 2003].

The position sensors (Hall effect sensors) send out a signal in the form of a three digit number which changes every 60° (electrical degrees).

Belt gear

The motors are not explicitly driven by the motor's output shaft, but are connected with a belt gear between the shaft and the pan- and tilt axes, respectively. The belt gear adds more dynamics to the system, and results in a greater torque, which is transferred from the driving wheel (motor's output shaft) to the pan- or tilt axis according to the following expression:

$$T = (F_2 - F_1)r_2$$

where T denotes the torque of the driven wheel (Nm), F_1, F_2 denotes the forces on each side of the belt (N), and r_2 denotes the radius of the driven wheel (m) [ToolBox, 2008]. The angular velocity is transferred by 6.1,

$$\omega_2 = \frac{r_1}{r_2}\omega_1 \tag{6.1}$$

where ω_1, ω_2 denotes the angular velocity for the driving wheel and the driven wheel, respectively. r_1 denotes the radius of the driving wheel [Plantenberg and Hill, 2013]. Naturally, the belt has some elasticity, which could affect the dynamics of the system.

6.2 Mathematical modeling

Much research was conducted in order to find a way to derive a mathematical model and, furthermore, a transfer function for the motors in the positioning unit. The idea was to control the velocities of the motors and in that way counteract for the external influences, ultimately stabilizing the positioning unit. Therefore, the desired mathematical model should take the motors' operational characteristics into consideration. Most alternatives found on how to derive a mathematical model for the motors gave a resulting transfer function based on the relation between the angular velocity and the DC-bus voltage of the motor. Due to the built-in motion controllers, which are connected to the each motor, this was not a suitable solution. A more thorough description of these motion controllers and their drawbacks is presented in following subsection.

Motion controllers

The built in motion controllers in the positioning unit that internally control the velocity and acceleration of the pan- and tilt motors follow a so-called S curve model. The S curve model is a type of motion trajectory planning, that can be divided into seven individual steps [Lewin, n.d.]. As seen in Fig. 6.2 the acceleration starts (phase I) with a linear increase, until it reaches the maximum acceleration, and keeps that acceleration in phase II until the maximum velocity is reached. The acceleration then decreases in phase III until it reaches zero, and in phase IV it starts to decelerate. The behaviour in phase IV - VII is symmetric to the behaviour in the first three phases. To determine on which form the S curve should be designed, the mechanical nature of the system has to be known [Lewin, n.d.]. For example, to get a smoother movement, phase II and phase VI could be removed. There is a way to mathematically determine the profile of the S curve, though this is not always intuitive due to the possible model variations.

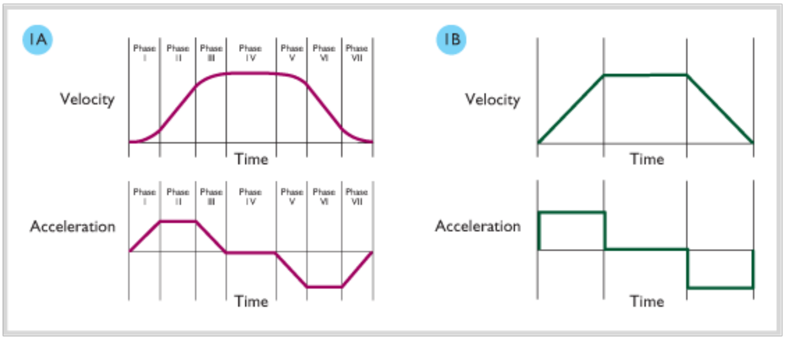


Figure 6.2: The seven phases of an S curve profile [Lewin, n.d.].

In discrete time, the acceleration, velocity and position could be expressed as in 6.2 to 6.4,

$$P_t = P_t + V_t + \frac{1}{2}A_t + \frac{1}{6}J \tag{6.2}$$

$$V_t = V_t + A_t + \frac{1}{2}J_t \tag{6.3}$$

$$A_t = A_t + J_t \tag{6.4}$$

where P_t, V_t and A_t are the position, velocity and acceleration at time T and J is the jerk profile (time rate of acceleration) [Lewin, n.d.]. The idea was to incorporate this model in the derivation of the mathematical modeling of the motors, to enhance the credibility of the identified model. During this research, it was found that the motion controller was updated dynamically in order to prevent any motions of the motors exceeding any preset acceleration- or velocity limits of the unit. This would mean that several mathematical models of the motors would have to be derived, one for each change in velocity. The approach of deriving this type of mathematical model was therefore rejected.

Belt gear oscillations

Fig. 6.3 shows a generated step in the motors (green), which result is read from the IMU in the camera housing (blue), which is at the end of the tilt axis. It is clearly seen that there is a somewhat oscillating behaviour, probably partly caused by the elasticity in the belt, but could also be the result of an unsteady fastening of the camera (holding the IMU). In order to get rid of these oscillations, this behaviour would have to be modeled and some damping coefficient introduced. This was considered in the modeling of the motors (red line in Fig. 6.3), but the conclusion was that the effort needed to model these oscillations in a reliable way was too large considering the scope of the thesis and the available amount of time. What caused the oscillations was also considered too uncertain to establish, which was another reason for not putting more effort into the problem, and instead an alternative way to derive a suitable model for the motors was investigated. Also, the oscillations could disturb the image, but was considered small enough to be handled by the EIS.

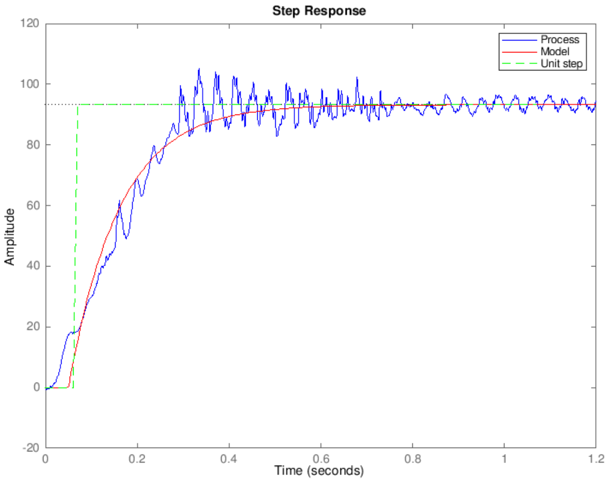


Figure 6.3: Disturbances due to belt gear elasticity, amongst others.

6.3 System identification

System identification is an approach that deals with the problem of creating a dynamic model of a system. The main purpose is to investigate the relation between input- and output signals. There are several different methods comprised by the term system identification, with different levels of complexity. When little or no knowledge exists about a system, a method called black-box model could be used, [Ljung, 2007], which was applied in this thesis.

A benefit of a black-box model is that, in contrast to a physical model, it only describes the relation between the input- and output signals of the system and does not take any underlying physics into consideration [Ljung and Glad, 2003]. As for system identification, there are several methods for creating black-box models. The method investigated in this thesis is presented below.

Step response identification

Analyzing step responses is a simple yet frequently used method for determining how a system behaves. The basic idea of this method is that the input signal is changed to a constant value while studying the output signal to see how it reacts to the change [Ljung and Glad, 2003]. The input signal, $u(t)$, is changed as described in 6.5.

$$u(t) = \begin{cases} u_0 & ; \quad t < t_0 \\ u_1 & ; \quad t \geq t_0 \end{cases} \tag{6.5}$$

The change in the input signal should optimally result in an output signal in the shape of a step. Depending on the actual result compared to the optimal result, it is possible to study the output signal's behavior and investigate eventual time delays, static gain and whether the response is, for example, damped or oscillatory.

The step response identification performed in this thesis was carried out on the pan motor. Since both motors were of the same type, it was assumed that the tilt motor would behave similarly. The input signal that was sent in to the system was a velocity command to the motor in degrees per second ($^\circ/s$). The output signal was then generated in two different ways. In the first one the resulting pan velocity (= output signal) was read from the IMU in the camera, since there already existed a software tool to read and plot the measurements in real time over a UDP socket, which allowed for smooth communication. The other way was to implement a small code snippet that enabled reading the velocity directly from the out axis in the pan motor. The IMU placement in the housing was not as robust and more sensitive to vibrations than the latter alternative, however, reading the velocity from the motor axis does not take into account the elasticity in the belt gear, which could also yield misleading results. Despite of this, it was argued that the latter alternative was the most robust way to perform the system identification, given the available time, and the assumption that the elasticity would yield errors in a range that could be handled by the EIS.

A few different step responses were created with different input signals. The velocity changes that were chosen as input signals to generate a step response are shown in Table 6.1.

Starting velocity ($^\circ/s$)	**Final velocity (step input signal) ($^\circ/s$)**
0	30
0	60
0	90
0	120
30	60
60	90
90	120

Table 6.1: Input signals used to create the step responses.

Results from step response identification

The results from the step response identification showed that the system behaved differently at different velocities, thus no single transfer function could be found to describe all step responses. This could be an indicator that the system is of nonlinear character [Ljung and Glad, 2000], or that the system identification is performed without taking the velocity trajectory (S curve) into consideration. Fig. 6.4 shows two suggestions of how to perform the black-box system identification. The upper red arrows shows the way in which the first tests were conducted, by having the command velocity as input signal and the actual motor velocity as output signal. This means that no consideration is taken to the S curve generation since this block lies between the input and the output, thus the black-box model will not take these dynamics into account. Another option would be to perform the system identification having the output from the S curve as an input to the black-box and the same output. In this way the behaviour of the motor could be isolated and a linear system that fits a single transfer function could most likely be found.

Due to lack of time and knowledge about the existing implementation of the trajectory generation, this alternative had to be disregarded. Figs. 6.5 to 6.7 show the results from the system identification where the velocity was read from the motor axis. As can be seen in these plots, the slope of the step varies with the change in velocity. For a smaller step in velocity, the slope is steeper.

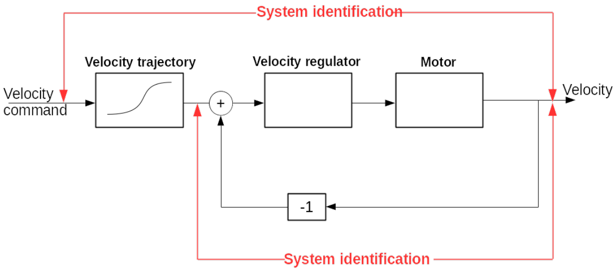


Figure 6.4: System identification options.

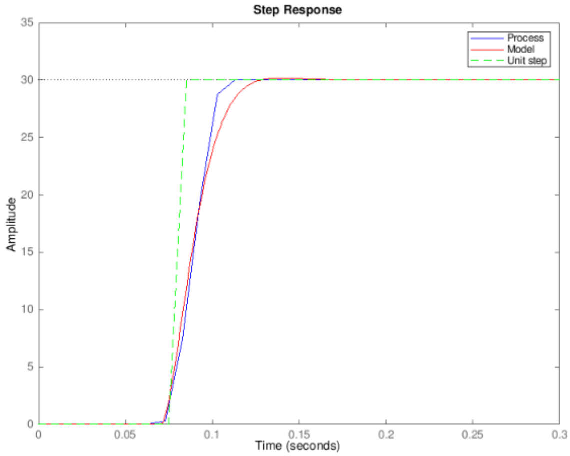


Figure 6.5: System identification for a velocity change of 30°/s.

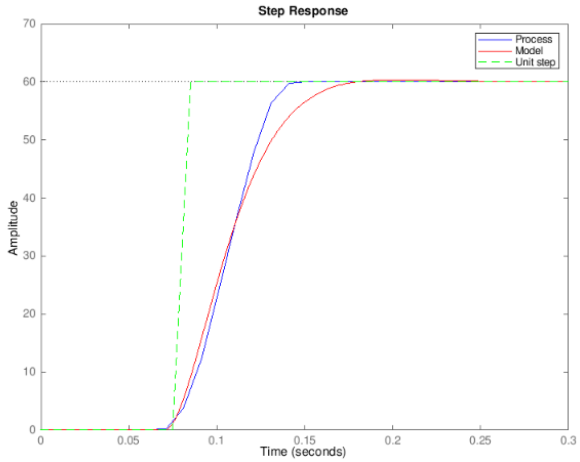


Figure 6.6: System identification for a velocity change of 60°/s.

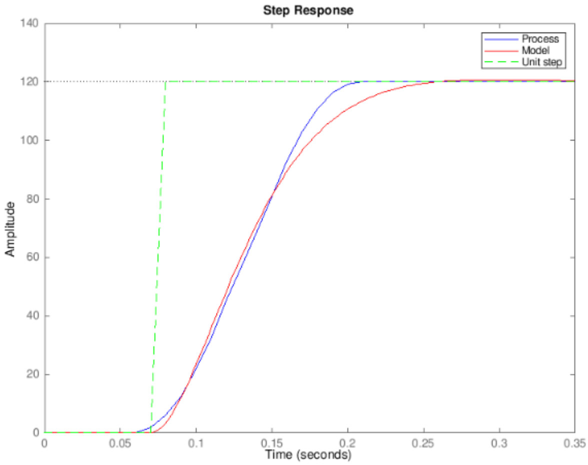


Figure 6.7: System identification for a velocity change of 120°/s.

The overall conclusion from the system identification was that the motors behaved in accordance with a second order system on the form

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} e^{-Ls}$$

with the same delay L and damping ratio ζ, whereas the natural frequency ω_n varies with velocity. These values were found empirically, with results shown in Table. 6.2.

Velocity ($^\circ/s$)	L	ζ	ω_n
30	0.07	0.85	85
60	0.07	0.85	45
90	0.07	0.85	29
120	0.07	0.85	26

Table 6.2: Parameters derived from empirical experiments.

Since no single transfer function was found describing the system, an alternate solution of finding a suitable model for describing the system had to be found. One option was to find a "mean value" of the different transfer functions derived, to describe most of the behaviours equally well, even though it would not be optimal. Another option was to design a set of transfer functions, each one customized to a certain range of velocities. In this way each step response would yield a specific transfer function and, further on, individual controllers. This would be a simple way to solve the problem with varying behaviours, but would require more computational power and a more complex implementation. After consulting the supervisor at Axis, the computational power was said not to become a problem, thus both alternatives were tested.

7

Control

This section includes the theory and methodology behind the design of the control system. Most of the theory investigated in this section is only briefly presented. For a deeper understanding of the concepts, the reader is referred to the external references stated in the bibliography.

7.1 Controller

After finding suitable transfer function(s) to describe the motors, a controller was to be chosen and designed. Below, the controllers investigated for the purpose of this thesis are presented. The general idea was that two controllers should be designed, one for the pan motor and one for the tilt motor, respectively. As stated in Section 1.3, it was assumed that the motors could be controlled independently of each other. Fig. 7.1 shows the graphical representation of the proposed control concept,

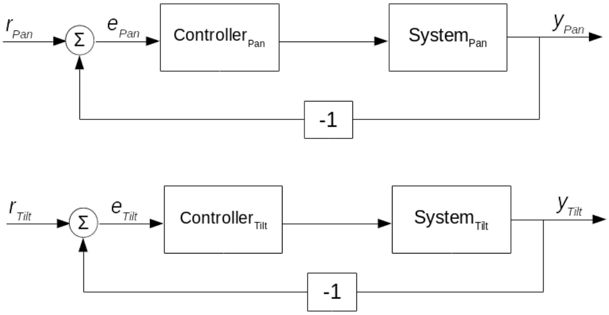


Figure 7.1: Control concept.

where r_{Pan} and r_{tilt} are the reference signals, which should represent the desired position to which the housing should move. e_{Pan} and e_{tilt} are the errors between the desired position and the current position, and y_{Pan} and y_{Tilt} are the output signals from the system, representing the encoder values of each motor, working as the feedback loops.

PID controller

After studying similar published works within gyro stabilization, it was concluded that either a PI- or a PID controller would be a good fit for this type of system. A classic PID controller could be written on the form:

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de}{dt} \tag{7.1}$$

where u is the control signal and e is the control error. k_p, k_i and k_d describe the proportional, integral and derivative parts, respectively [Åström and Murray, 2012a]. If a PI controller were to be used, the k_d term would simply be set to zero. The transfer function for a PID controller can be created from 7.1, which results in 7.2 [Åström and Murray, 2012b],

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s \tag{7.2}$$

where s is the complex frequency. Just like in 7.1, the K_d term is set to zero when using a PI controller.

An issue that can occur when using a PI- or PID controller is that the integral term of the controller, which works as an accumulation of past errors, performs the accumulations at a high pace. This can then lead to overshoots in the process output and could cause the actuator to saturate. This phenomenon is called integrator windup and is something that needs to be taken into consideration. Integrator windup can be avoided by implementing an anti-windup method [Åström and Murray, 2012a]. The method implemented to avoiding integrator windup in this thesis is called tracking. A block schedule of a PID controller with tracking is shown in Fig. 7.2. The main idea of this method is to introduce an extra feedback path that measures the difference (error) between the calculated control signal and the actual one. This error is denoted as e_s and is fed into the integrator which will force the integrator output towards a specific value, leading to the integrator value becoming zero. This will only occur when the actuator is saturated, i.e. when $v \neq u$. The gain in Fig. 7.2 denoted as k_t is called the tracking time constant. This gain decides how fast the integrator output will converge towards the specific value described above [Årzén, 2019].

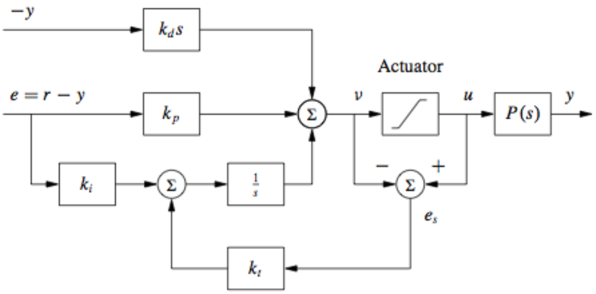


Figure 7.2: Block schedule of a PID-controller with tracking [Åström and Murray, 2012a].

The way a PI- or PID controller is normally implemented in processes differs in some ways to how the theoretical controller is presented and described. The way it differs is how the set-point and the measured value are treated. In the original controller, these values are treated the same way while in most cases of implementation, they are treated differently for improved performance. This means that a smaller fraction of the set-point value is used in the proportional and derivative term. This method is called set-point weighting and results in the modified controller, in the frequency domain, described in 7.3,

$$U(s) = K_p \cdot (\beta \cdot R(s) - Y(s) + \frac{K_i}{s} \cdot (R(s) - Y(s)) + K_d \cdot s \cdot (\gamma \cdot R(s) - Y(s)) \tag{7.3}$$

where β and γ are the set-point weights and $R(s)$ and $Y(s)$ are the set-point and the measured output, respectively [Årzén, 2019].

Discretization

As well as for the set-point weighting, discretization of the controller is something that needs to be done in order to implement the controller in the software. The discretization is performed on all three terms of the controller. However, the proportional term can be implemented as it is, while the integral- and derivative term need to be altered. There are different methods of doing this, including forward- and backward Euler approximation. In the frequency domain, these methods are described as in 7.4 and 7.5, respectively,

$$s \approx \frac{z-1}{h} \tag{7.4}$$

$$s \approx \frac{z-1}{zh} \tag{7.5}$$

where h is the sampling interval of the control algorithm. The sampling time for this system was set to $h = 50ms$, since this is a commonly used number and also seemed reasonable considering the amount of calculations that should be executed periodically, together with the computational power available. The idea is to use forward approximation on the integral term and backward approximation on the derivative term. The resulting control law is then described as in 7.6,

$$U(z) = K_p \cdot (\beta \cdot R(z) - Y(z) + \frac{K_i \cdot h}{z-1} \cdot (R(z) - Y(z)) + K_d \cdot \frac{z-1}{z \cdot h} \cdot (\gamma \cdot R(z) - Y(z)) \tag{7.6}$$

which is suitable for implementation in the system software [Årzén, 2019].

Control design and simulation

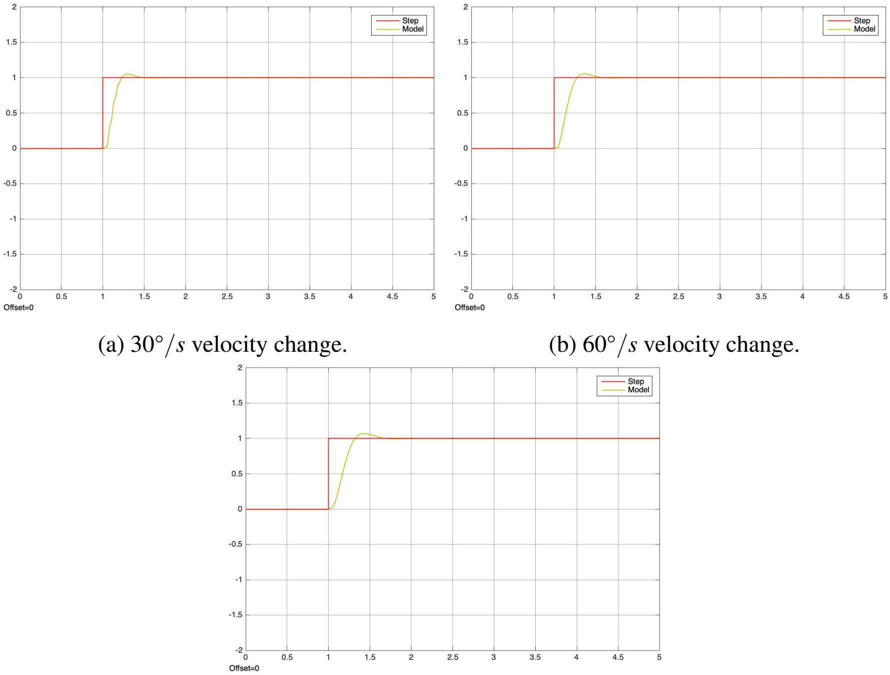
The controller parameters were determined based on the motors' transfer functions. Since the system identification resulted in four different transfer functions, four different controllers were designed. To determine the parameters, all parameters of the motors' transfer functions, including the time delay, were taken into account. Several simulations of the closed loop response were carried out using Simulink and resulted in the choice of using a PI controller for both the pan- and the tilt motor, respectively. The choice of using this type of controller fell on the fact that the simulations showed useful and suitable results for all velocity changes.

The controller could then be derived empirically. The resulting design of the PI controller for the various transfer functions as well as for a general transfer function (mean value) can be seen in Table 7.1.

Velocity ($^\circ/s$)	K_p	K_i
$\leq$ **30**	0.019	7.587
$\leq$ **60**	0.035	6.671
$\leq$ **90**	0.060	5.858
$\leq$ **120**	0.065	5.658
General	0.045	6.844

Table 7.1: PI-parameters for the different transfer functions.

After determining the parameters, the controllers were discretized and closed loop systems were formed. Using these controllers, the closed loop systems could be simulated, using a step reference signal. The closed loop step responses, for the various transfer functions and the general one, were simulated in discrete time and the results can be seen in Figs. 7.3 and 7.4 which show that there is a small overshoot in the step response for all velocity changes. This was however considered acceptable due to the trade off between a shorter rise time and a larger overshoot. It can also be seen that the error quickly decreases towards zero, which is preferable.

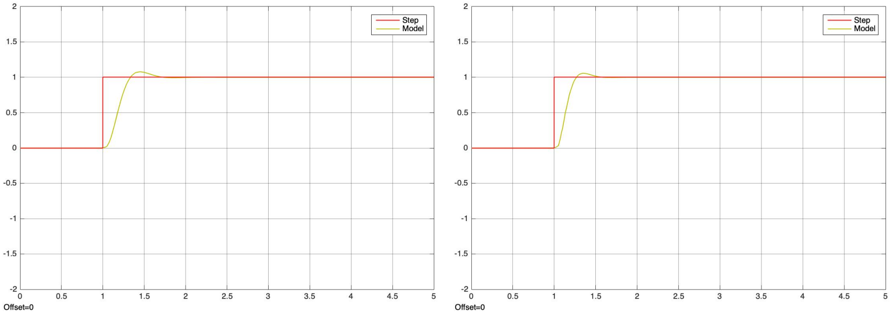


(a) 30°/s velocity change.

(b) 60°/s velocity change.

(c) 90°/s velocity change.

Figure 7.3: Simulated closed loop step response discrete time for 30, 60 and 90 °/s velocity change.



(a) 120°/s velocity change.

(b) General velocity change.

Figure 7.4: Simulated closed loop step response in discrete time for 120°/s and general velocity change.

The closed loop systems were also simulated with reference signals in the shape of a ramp, a triangular wave and sinusoidal waves at various frequencies. These simulations gave similar results and showed that the systems were stable. These results were therefore not included in the thesis.

7.2 Sensor fusion

Sensor fusion is a term explaining the combination of sensory data in order to enhance the performance of the sensors and the resulting data [Kopetz, 2002]. As previously mentioned in Section 2.1, the sensors used in this thesis are sensitive when used independently. The accelerometer is sensitive to vibration and mechanical noise, and the gyroscope is sensitive to drift [SparkFun Electronics, 2013a], thus some kind of sensor fusion is needed to enhance the performance of the IMU. The angular estimations from the gyroscope and the accelerometer derived individually showing the drawbacks of both sensors are shown in Fig. 7.5 and 7.6. From Fig. 7.5 the integrated gyroscope measurements are shown, visualizing the drift. Both the roll, pitch and yaw angles starts and ends with an angle of zero in reality, but the figure of the measured values it shows that the final angles are far from zero, e.g. approx. 25° in roll and approx. −40° in pitch. Thus it is concluded that drift has occurred in the gyroscope. In Fig. 7.6, the angle estimates from the accelerometer readings are shown, which are clearly a bit noisy compared to the angle estimates from the gyroscope. Considering these individual drawbacks, a sensor fusion method enhancing the behaviour in both sensors was desired.

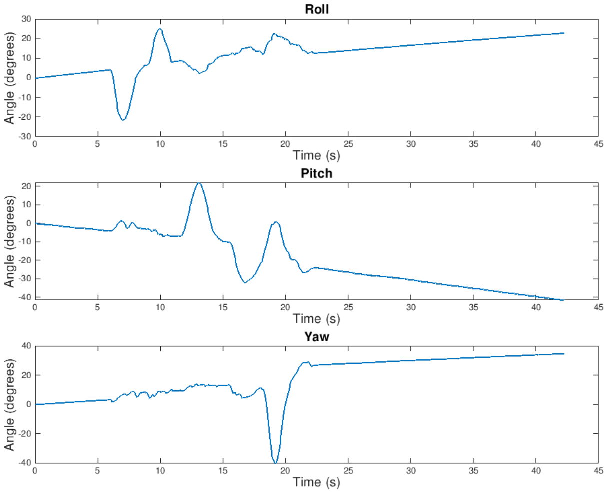


Figure 7.5: Pitch and roll angles integrated from the gyroscope.

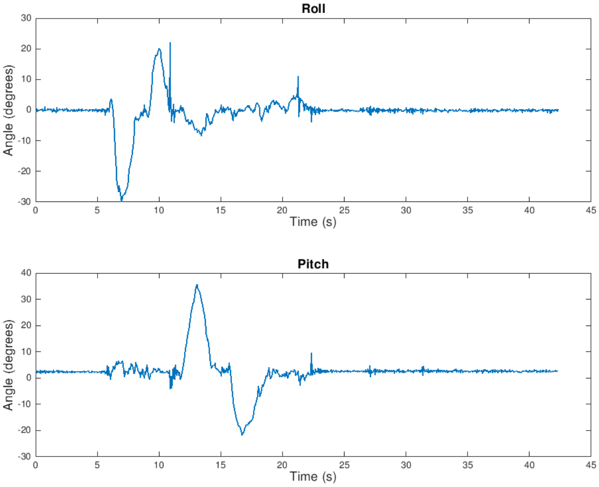


Figure 7.6: Pitch and roll angle estimation from the accelerometer.

Kalman filtering

A Kalman filter is an observer that can be used for noisy measurements. The aim of implementing the observer is to get an estimate of the states of the system so that a cost function (mean squared error) could be minimized [Åström and Murray, 2018]. Considering a dynamic system, it can be expressed in discrete time state space form as,

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t$$
$$\mathbf{y}_{t+1} = C\mathbf{x}_{t+1} + \mathbf{w}_{t+1}$$

where A, B and C are the system matrices and $\mathbf{v}_t$ and $\mathbf{w}_{t+1}$ are the process and measurement noises, respectively [Salmony, n.d.]. The dynamic system considered in this application can be described as in 7.7 for the pitch angle, which is equivalent for the roll and yaw angles.

$$\theta_t = \theta_{t-1} + (\dot{\theta}_t - b_{t-1})h \tag{7.7}$$

θ_t denotes the current pitch angle measured by the gyroscope, θ_{t-1} the previous pitch angle, $\dot{\theta}_k$ the current gyroscope rate, b_{t-1} the bias from the previous measurement and h the sampling rate [Gustavi and Andersson, 2017]. This system can be expressed on state space form as in 7.8 and 7.9 below,

$$\dot{x}_t = \begin{bmatrix} \hat{\theta}_t \\ b_t \end{bmatrix} = \begin{bmatrix} 1 & -h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_{t-1} \\ b_{t-1} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \dot{\theta}_k + w_k \tag{7.8}$$

$$y_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \dot{x}_t + v_t \tag{7.9}$$

where the state vector of the system is the angle estimate and the bias. The measurement parameter can be expressed as,

$$z_t = \hat{\theta}_{Acc}$$

where z_t is the angle estimate computed from the accelerometer readings [Salmony, n.d.]. The Kalman filter is a recursive filter and consists of two parts; prediction and correction. The predictive part computes both the estimate and the error, which are then both corrected in the correction part, using the measurement [Åström and Murray, 2018]. The recursive process looks as follows, starting with the prediction part,

$$\begin{aligned} \dot{x}_{t+1} &= Ax_t + Bu_t \\ P &= APA^T + Q \end{aligned}$$

where P denotes the error covariance matrix and Q the process noise covariance matrix explaining the certainty of the dynamical model used [Salmony, n.d.], which therefore has to be derived empirically. Q is a diagonal matrix as in 7.10, where it was assumed that the noise is uncorrelated [Åström and Murray, 2018].

$$Q = \begin{bmatrix} Q_\theta & 0 \\ 0 & Q_b \end{bmatrix} \tag{7.10}$$

The correction phase of the filter follows the prediction part and is computed the following way,

$$\begin{aligned} \tilde{y}_{t+1} &= z_{t+1} - C\dot{x}_{t+1} \\ S &= CPC^T + R \\ K &= PC^T S^{-1} \\ \dot{x}_{t+1} &= \dot{x}_{t+1} + K\tilde{y}_{t+1} \\ P &= (I - KC)P \end{aligned}$$

where S is the error covariance matrix of the prediction, adding R, which is the measurement noise covariance matrix, which depends on the specific sensors used, and can therefore be found using a data sheet [Salmony, n.d.]. K is known as the Kalman gain, and is optimally chosen depending on both the noise processes and the dynamics of the system, and is constant when the process has converged [Åström and Murray, 2018].

Unfortunately, since the accelerometer cannot measure the yaw angle, the Kalman filter cannot correct this angle, which has to be derived using only the gyroscope measurements. This is done by integrating the measured yaw rate. Fig. 7.7 shows the result of the Kalman filter for both the roll- and pitch angles based on the IMU readings shown above in Figs. 7.5 and 7.6. The figure also shows the unfiltered gyroscope- and accelerometer measurements, respectively.

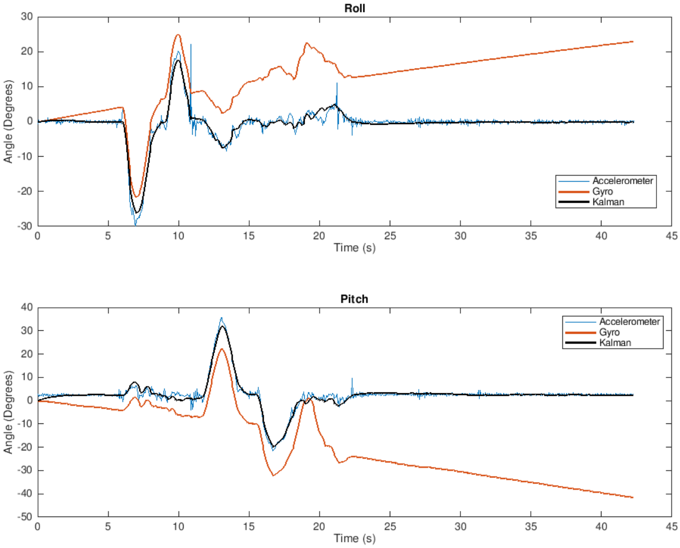


Figure 7.7: Roll- and pitch angles as a result of the Kalman filter.

The figure clearly shows that the filtered measurements are not affected by the drift from the gyroscope or the sensitivity to noise from the accelerometer. The results and the sensor fusion as a whole are therefore considered desirable.

Alternative filtering methods

Alternative filtering methods for the IMU were considered for this thesis, such as the Complementary filter. These alternatives were disregarded without further investigation since previous similar projects, e.g. [Gustavi and Andersson, 2017], have shown that the Kalman filter is enough to prove the concept of this thesis.

7.3 Reference creation

In order to control the pan- and tilt movements and counteract for the disturbances, some kind of reference signal to the control system has to be created. By knowing how the positioning unit is oriented relative its desired position (which is set by the operator), and in that way knowing the displacement (error) from this position, this reference signal could be derived. The relation between the coordinate systems is presented using Euler angles as described in section 5.1.

Knowledge about the positioning unit's current orientation is concluded from the information given by the pan- and tilt encoders together with the measurements from the IMU. The part of the positioning unit where the disturbances enter is also of interest to the orientation algorithm. In this case, the chassis of the positioning unit is fixated onto the desired object the positioning unit is mounted to, and will therefore be subject to the disturbances.

The sought-for result from the Euler angle representation is hence the Euler angles describing the rotation from the current position of the chassis needed to reach the desired position of the unit. In order to reach this representation, a set of matrix multiplications has to be conducted relating the interesting coordinate systems individually. The rotation matrices of interest are presented below:

$R_{chassis}^{IMU}$ - Rotation matrix describing the orientation of the chassis relative to the IMU, which is only a pan rotation, given by the pan encoder. This matrix could be created at any time, since this information could always be retrieved from the pan encoder.

R_{IMU}^{NED} - Rotation matrix describing the orientation of the IMU relative to the NED coordinate system. This rotation matrix could also be retrieved at all times using the measurements from the IMU and inserting them into 5.4.

$R_{NED}^{desired}$ - Rotation matrix describing the rotations in NED relative to the desired orientation of the housing. The desired orientation of the housing is set by the operator, and is expressed in the NED coordinate system. Since this is an arbitrary set orientation, this rotation matrix could always be created. Here, the roll angle always will be set to zero since the positioning unit cannot counteract for motions in roll.

$R_{chassis}^{desired}$ - The rotation matrix describing the relation between the chassis and the desired position of the housing.

Knowing these rotation matrices, relation 7.11 could be derived. Since all the matrices on the right-hand side of the equation could be created at all times, the chassis orientation relative to the desired orientation could always be found.

$$R_{chassis}^{desired} = R_{NED}^{desired} R_{IMU}^{NED} R_{chassis}^{IMU} \tag{7.11}$$

Fig. 7.8 shows the coordinate systems of interest, and their respective rotation axes. From this figure it can be seen that it differs 180° between the IMU coordinate system and the chassis and NED coordinate systems. The coordinate system in which the desired position is represented is not shown in this figure, since this coordinate system coincides with NED.

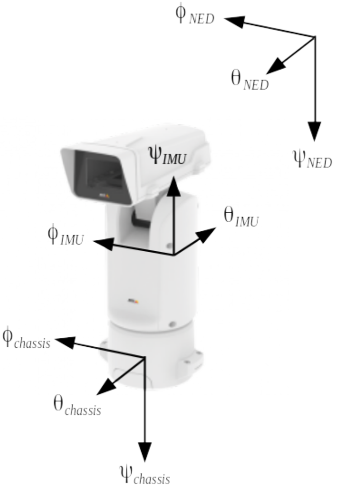


Figure 7.8: Coordinate system representation in the positioning unit.

The corresponding Euler angles could then be extracted from this final rotation matrix, using 5.6 to 5.8, yielding the angles to which the pan- and tilt motors should be set to rotate in order to counteract for the disturbances. By simplifying 7.11, it can be written as 7.12, and the relations 5.6 to 5.8 could be then be expressed as below, and the desired reference angles obtained.

$$R_{chassis}^{desired} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{7.12}$$

$$\phi_{ref} = atan2(-R_{32}, R_{33})$$

$$\theta_{ref} = atan2(R_{13}, \sqrt{(}R_{11}^2 + R_{12}^2))$$

$$\psi_{ref} = atan2(sin\phi_{ref}R_{31} - cos\phi_{ref}R_{21}, cos\phi_{ref}R_{22} - sin\phi_{ref}R_{32})$$

Figs. 7.9 to 7.11 show the created reference signals for the pitch angle when the input disturbance is a sinusoidal wave, a triangular wave and a unit step. As can be seen in the figures, the created reference signals follow the input disturbance, but with opposite sign. This means, theoretically speaking, that the reference signals will counteract the input disturbances. The same references are created for roll and yaw, respectively, though these plots are not shown here.

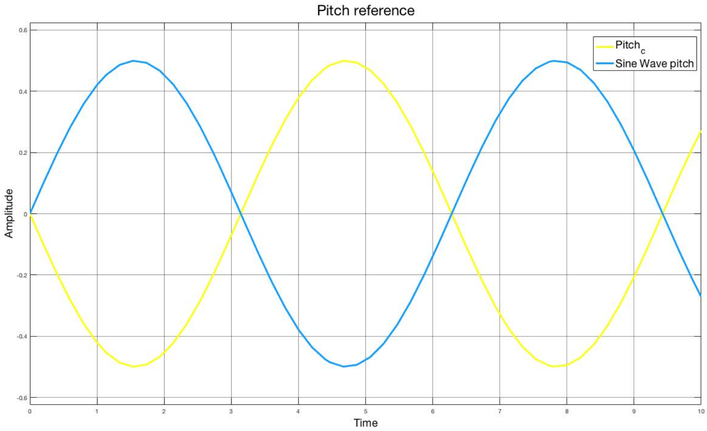


Figure 7.9: Sine wave reference signal for pitch.

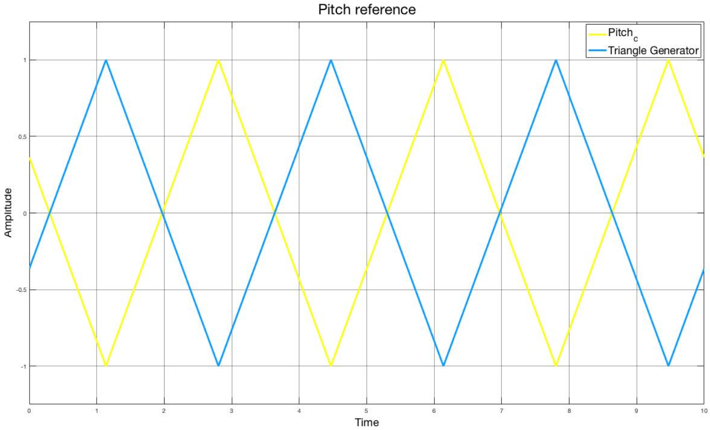


Figure 7.10: Triangle wave reference signal for pitch.

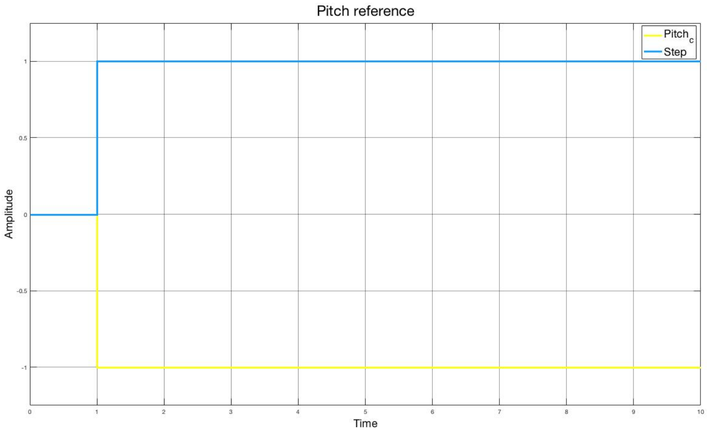


Figure 7.11: Unit step reference signal for pitch.

7.4 Compensating for roll

Since there are only motors for pan- and tilt movements, disturbances that yields a roll angle are not intuitive to counteract for. An idea emerged during the process, which intended to partition the roll angle to be able to express it in pitch (tilt) and yaw (pan) angles, and in that way be able to counteract for roll as well. Fig. 7.12 shows how a change in the roll angle would affect the pitch and yaw axes, respectively. An angular change α would result in the same angular difference in the other axes. This change would furthermore result in a displacement of these axes, dx and dy, which could be counteracted by the pan and tilt motors. These displacements could be calculated as in 7.13 and 7.14 where the length of the vectors corresponds to the distance measured by the accelerometer in the same axis.

$$dx = tan\alpha|accel_x| \tag{7.13}$$

$$dy = tan\alpha|accel_y| \tag{7.14}$$

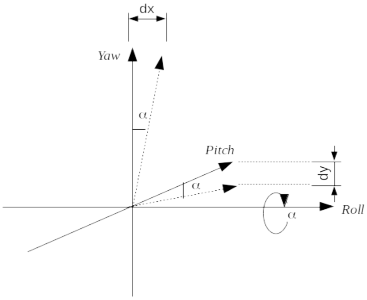


Figure 7.12: Angular change in roll.

The idea was to incorporate these calculations into the implementation of the control system, in order for the pan- and tilt motors to partly compensate for roll as well. This idea was later disregarded due to lack of time and conviction that the method would work.

8

Implementation

This chapter presents the creation of the stabilization algorithm, where all previously presented theory and sub-parts are put together to form the final algorithm. The aspects of implementation of the stabilization algorithm in the positioning unit software are also presented briefly.

8.1 System design

With all necessary parts of the stabilization algorithm defined individually, the whole system could be put together for implementation, as seen in Fig. 8.1. Data from the IMU is read and processed according to the aforementioned methods. The gyroscope- and accelerometer data are transformed according to section 5.2 and inserted into the Kalman filter, while the yaw rate of the gyroscope is integrated, giving ψ, and inserted directly into the Reference Creation block. Also inserted into the Reference Creation block are the resulting angles from the Kalman filter, θ, ϕ, the desired position of the positioning unit set by the operator, ϕ_d, θ_d, ψ_d and the current pan position of the unit, i.e. the pan encoder value $\psi_{encoder}$. The result from this block gives a pan- and tilt reference, as angles in radians, which are sent to the PI controllers. The PI controllers calculate the error between the current angle positions and the reference angle positions, e, and controls the velocity of the motors to minimize the error. This velocity is then sent to the motors in order to start a continuous motion towards the desired position. This results in a new position, y_{pan} and y_{tilt}, given by the encoders, which is sent back as the feedback loop.

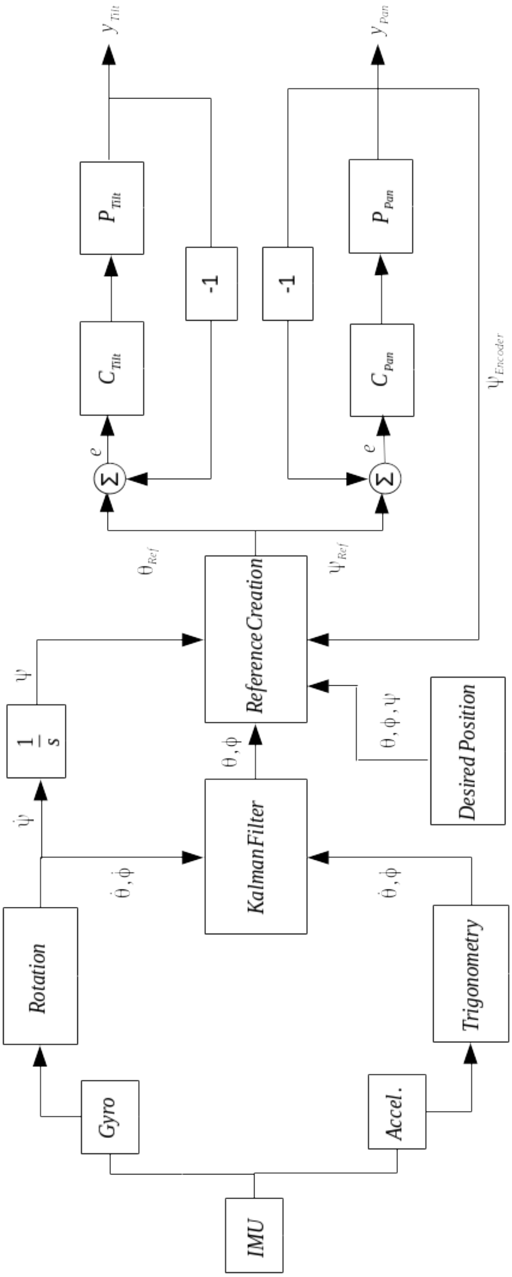


Figure 8.1: Schematics of the entire system.

8.2 Code structure

The stabilization algorithm was implemented in real-time in the main CPU in the positioning unit. The code was written in C and uploaded to the positioning unit via a JLink debugger from SEGGER [SEGGER, n.d.]. All blocks of the system in Fig. 8.1 were written as separate files, and then called from the main task, which executed the algorithm periodically with a cycle time of 50ms, to match the sample times of the Kalman filter and the PI controllers. All parameters were implemented as float variables, since there is an FPU in the MCU. Fig. 8.2 shows a graphical representation of the main task, including the external function calls. The first four blocks are only executed once, and handle the setting of constant parameters, communication setup and retrieving of the desired position from the operator. To simplify this step, the operator himself does not have to insert a desired position into the system. Instead, he directs the camera in the desired direction, e.g. using a joystick, and when the stabilization algorithm starts, the current encoder values will be set as the desired position. The command "stab on" indicates that the stabilization algorithm is started and the subsequent blocks are only executed if "stab on" is true. When the desired position is retrieved, the IMU is calibrated to remove the static offset. This is done by letting the positioning unit stay idle and taking 100 samples from the IMU, and then subtracting the mean value of these samples from the future measurements. This calibration is only executed once the first time "stab on" is issued, even if the same command is called several times while the code is running.

When these steps are completed, the stabilization algorithm as visualized in Fig. 8.2 is performed periodically. If the command "stab off" is issued, the stabilization algorithm stops, after a delay of $t = 100ms$, and the pan- and tilt motors are stopped. If the desired position needs to be changed during the course of the stabilization, the algorithm would have to be stopped, the new desired position set, and then the algorithm could be started again, with the new settings.

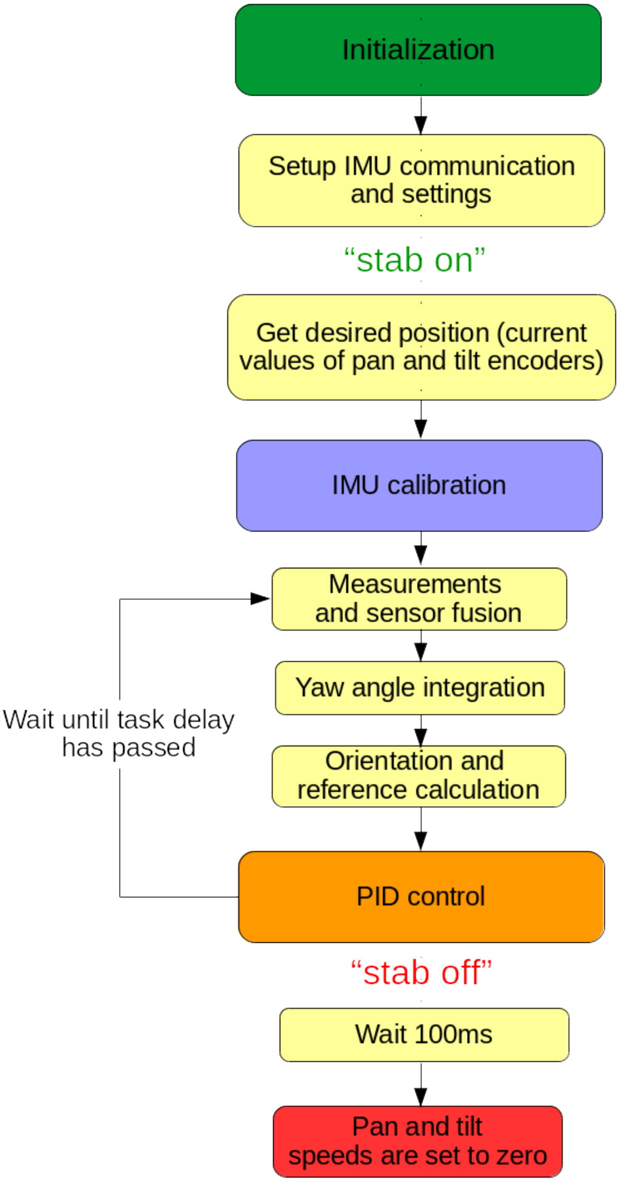

Initialization
Setup IMU communication and settings
“stab on”
Get desired position (current values of pan and tilt encoders)
IMU calibration
Measurements and sensor fusion
Yaw angle integration
Orientation and reference calculation
PID control
Wait until task delay has passed
“stab off”
Wait 100ms
Pan and tilt speeds are set to zero

Figure 8.2: Schematics representing the algorithm implementation.

Debugging

A fairly large amount of time was spent on debugging the written code. The code was written step-by-step and each part was tested and debugged independently. Thereafter, all parts were put together and the entire stabilization algorithm was tested.

To allow for smoother debugging and parameter tuning, a functionality was implemented to enable alterations of the controller parameters during run-time.

As a result from debugging the code, some of the sub-parts in the stabilization algorithm had to be altered from the original idea to work in the actual implementation. Most of the changes were insignificant enough not to be mentioned, whereas the changes made to the control algorithm are discussed in Section 8.3 below.

Also, during testing, an offset was found in the tilt system, where the "home position" according to the stabilization algorithm was at approx. 2.75°, rather than 0°. The reason for this was found most likely to be that the placement of the IMU is not fully aligned with the vertical axis, but with a small angular displacement of 2.75°. This displacement was added to the tilt encoder value in the start up phase of the stabilization algorithm, to eliminate its influence in the stabilization.

8.3 Controllers

After determining the controller parameters, the controllers was implemented and tested in the system software. Since the software already included a suitable controller, this was used rather than creating another file for the designed controllers. The tests that were carried out included introducing a reference signal in the shape of a step of a specific angle in order to calculate an output in the shape of an angular velocity which were to be sent to the pan- and tilt motors, respectively.

In theory, the behavior of the motors should have been similar to the simulations carried out in Simulink. However, this was not the case. Instead, the controller parameters were to be tuned once more in order to get desired results. The tuning of the parameters were done empirically for each motor since the results contradicted the assumption that the two motors would behave the same.

After discussing why the controller parameters differed from Simulink to the implementation in C code, a few conclusions could be drawn. One of these conclusions was that the way the calculations of the controller are done probably varies from Simulink to the existing code in the software. Another conclusion was the fact that the transfer functions obtained from the system identification could have been wrong. At this point though, the final controller parameters could be derived empirically which led to the conclusion that no search for new transfer functions was needed.

Various tests were conducted in order to find out the best control concept for to the stabilization algorithm. The concepts were to either have four different controllers for the different velocity changes or simply a general controller that would be appropriate for all kinds of velocity changes. Since the case with the four different controllers could introduce a small latency in the software, the computation time was studied during the tests as well as the speed and results of the controller.

The results of the tests showed that the best solution was to have four different controllers, per motor, for different velocities. This conclusion was drawn due to the fact that there was not any noticeable increase in the computation time and since the solution could be easily implemented in the software. The final empirical tuning of the different controllers' parameters resulted in new velocity intervals as well as the introduction of a derivative term in the tilt motor's controller. The four controllers were implemented in the software as a switch case, with four different if statements, depending on the velocity range. Inside the if statements the PID parameters were set, and the PID controller initialized. Thereafter, the PID algorithm was called, with the new parameter setting. This enabled a transition between the PID-controllers that did not result in a bump in the control signal. Since the if statements were explicitly defined, there was no risk of two if statements being true simultaneously.

The resulting controller parameters for the pan- and the tilt motor, as can be seen in table 8.1 and 8.2, were used in the final implementation, i.e. no consideration was taken to the aforementioned Simulink-designed parameters.

Velocity ($^\circ/s$)	K_p	K_i
$\leq$ **30**	10.0	7.0
$\leq$ **80**	9.0	4.0
$\leq$ **90**	8.0	0.7
$\leq$ **120**	7.0	1.0

Table 8.1: Final PI parameters for the different transfer functions (pan motor).

Velocity ($^\circ/s$)	K_p	K_i	K_d
$\leq$ **30**	10.0	7.0	0.0
$\leq$ **70**	7.0	0.8	0.2
$\leq$ **85**	7.0	1.0	15.0
$\leq$ **120**	5.0	15.0	30.0

Table 8.2: Final PI- and PID parameters for the different transfer functions (tilt motor).

As seen in the tables above, the controller parameters are not the same for the pan- and tilt motors, which was earlier expected. A reason for this could be that the motors are mounted in different ways, why they behave differently. Because of this, the gravitational force will act differently on the two motors, affecting their behaviour. Also, the powertrain is different for the two motors, which could also be a reason why the controller parameters differ. Furthermore, different powertrains yields different mechanical behaviours, which could also affect the choice of parameters, to compensate for a greater overshoot, or similar.

9

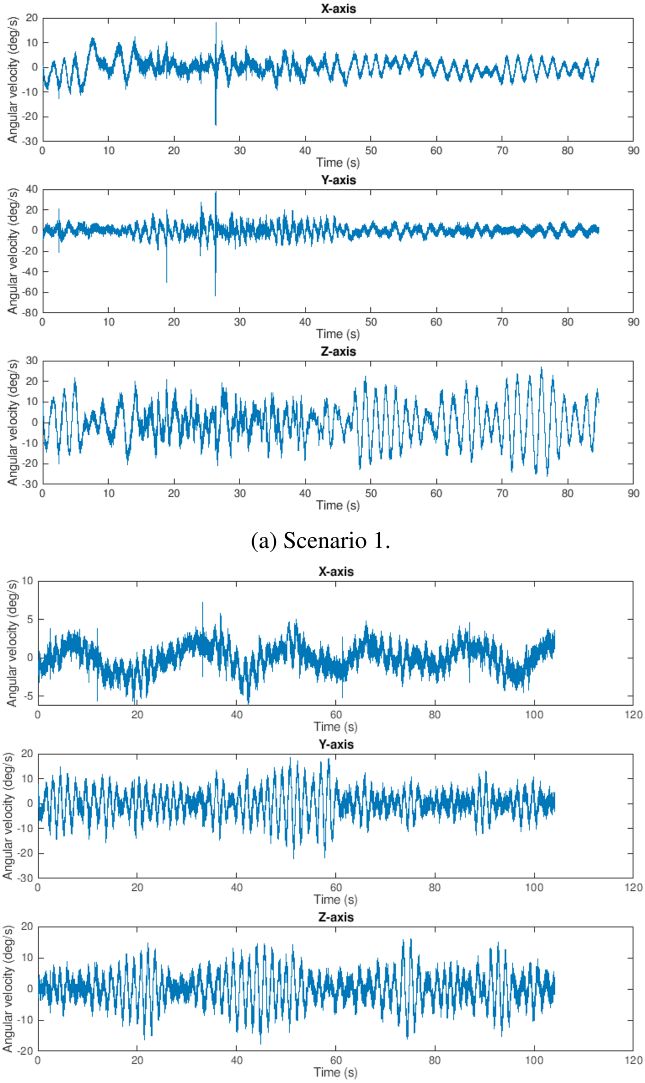
Experiments

Parallel to the implementation the stabilization algorithm, several experiments were performed. These were carried out in order to obtain valuable information about the requirements of the implementation, as well as performing functionality tests and simulations. The experiments are presented below.

9.1 Karlskrona data acquisition

In order to determine whether the accelerations, velocities and amplitudes in a real life scenario at sea exceed the limits of the positioning unit or not, various tests were performed. These tests were made possible thanks to the collaboration with the individual research program *Wallenberg AI, Autonomous Systems and Software Program (WASP)* [WASP, n.d.]. These tests were carried out in Karlskrona where different camera units, implemented with IMU:s, were placed on a boat. All together, two different scenarios were conducted and from which the IMU data was collected: one where the boat was stationary and affected by beam waves at 90° heading and one where it was affected by head waves at 0° heading.

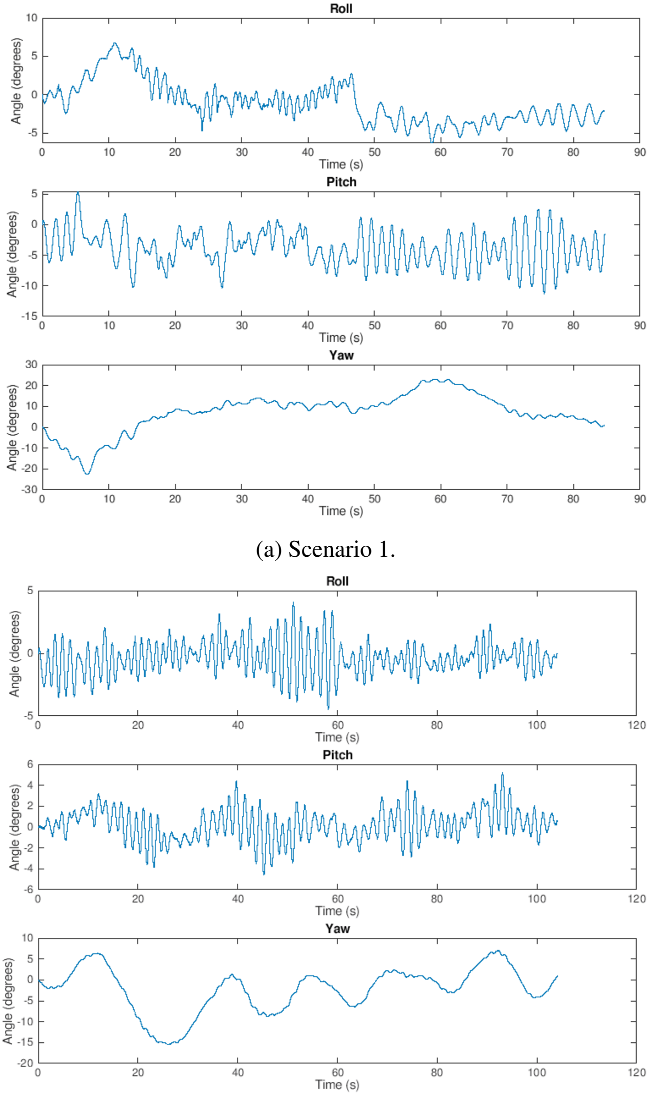
For both scenarios, the gyroscope- and the accelerometer readings were collected, processed and plotted in order to get a clear picture of how the real life scenarios would affect the positioning unit. The resulting plots show the gyroscope readings in $^\circ/s$ and the accelerometer readings in m/s^2, together with the derived and integrated gyroscope readings in $^\circ/s^2$ and °, respectively. The plots are presented in Figs. 9.1 to 9.4 in the same order as described. The IMU was installed in a way that the x-, y- and z axis of the IMU represents the yaw-, pitch- and roll axis, respectively.



(a) Scenario 1.

(b) Scenario 2.

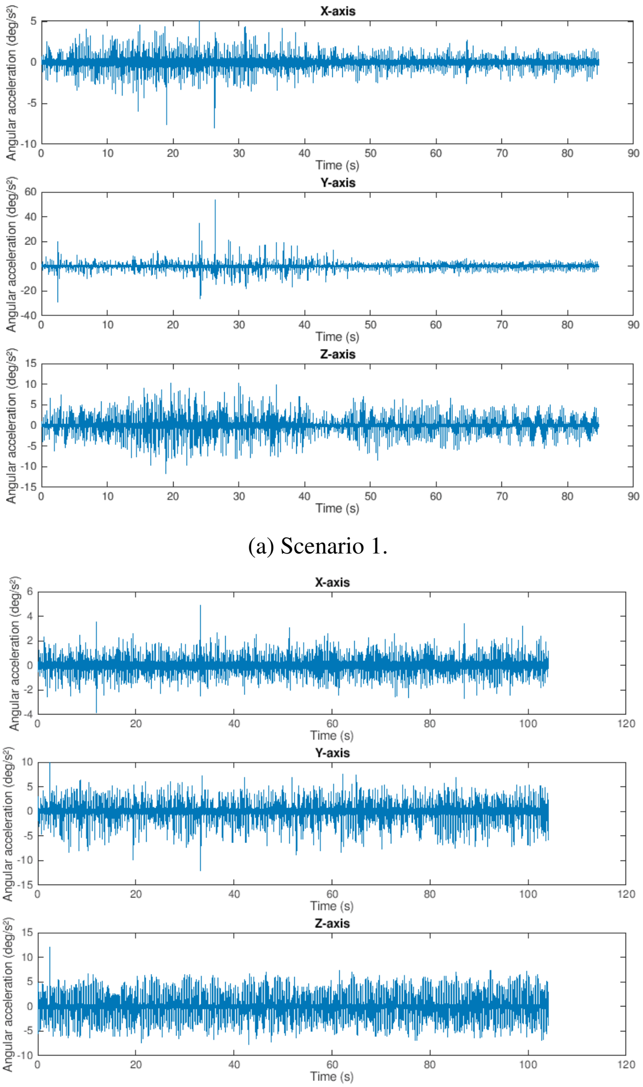
Figure 9.1: Gyroscope readings from both scenarios ($^{\circ}/s$).



(a) Scenario 1.

(b) Scenario 2.

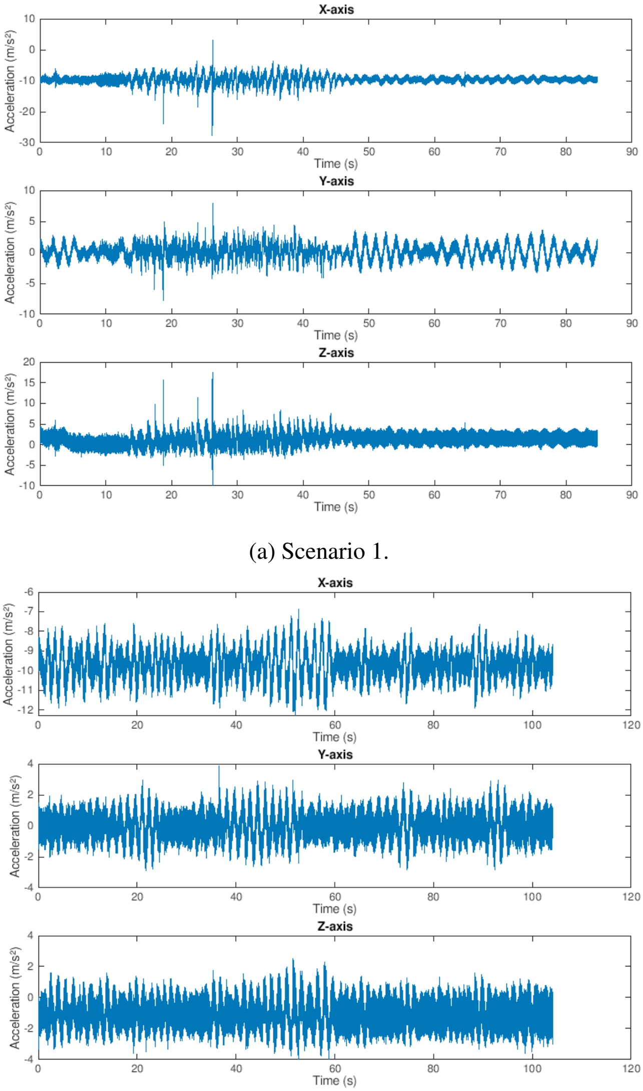
Figure 9.2: Integrated gyroscope readings from both scenarios (°).



(a) Scenario 1.

(b) Scenario 2.

Figure 9.3: Derived gyroscope readings from both scenarios ($^\circ/s^2$).



(a) Scenario 1.

(b) Scenario 2.

Figure 9.4: Accelerometer readings from both scenarios (m/s^2).

Although the readings presented above were not filtered, the readings gave an understanding of how large the external influences actually were and whether they exceed the limits of the positioning unit.

Both scenarios show that the angular velocity never exceeds $\pm 30^\circ/s$, which was within the limits of the two motors' velocities. The angle amplitude in both scenarios stay within $\pm 25^\circ$ in yaw and approximately $\pm 15^\circ$ in pitch which is within the limits of which the positioning unit can turn its motors in pan and tilt. The acceleration from the derived gyroscope data does not exceed $\pm 20^\circ/s^2$ besides from a few peaks at approximately $40 - 60^\circ/s^2$. All of these values, including the peaks, are within the limits of the motors' accelerations. The acceleration from the accelerometer also has a few peaks at approximately $\pm 25 m/s^2$ but does, otherwise, not exceed $\pm 15 m/s^2$. Since the motors' accelerations are measured in $^\circ/s^2$, there is no good way of converting those values to m/s^2. However, the values presented in the plot are not abnormal and are, therefore, considered applicable for gyro stabilization.

9.2 Robotlab

A few experiments were carried out in the Robotlab at LTH, in order to enable more thorough tests of the stabilization algorithm. The robot used for the experiments was an *ABB IRB 2400* [ABB Global, 2019] together with a custom-made mount for the tool changer and the positioning unit, designed and manufactured at Axis [Axis, 2019]. The *ABB IRB 2400* consists of 6 joints, and can handle a load of approx. 13 kg, why it was considered robust enough to carry the T99A10, which weighs approx. 11.5 kg. The resulting setup is shown in Fig. 9.5 where the positioning unit is locked in an upright position. The experiments that was carried out in Robotlab out was mainly testing of the implementation and further tuning of the controller parameters to obtain the best behaviour of the positioning unit. Furthermore, simulations were made for visualization of the stabilization algorithm together with movie recordings from the camera in parallel.

Pan and tilt were tested both individually and together, to ensure the system worked in all situations. Different joints of the *ABB IRB 2400* were jogged in order to get a wide range of simulations. Also, data was gathered from the simulations for plotting of the reference tracking and visualization of the control system.

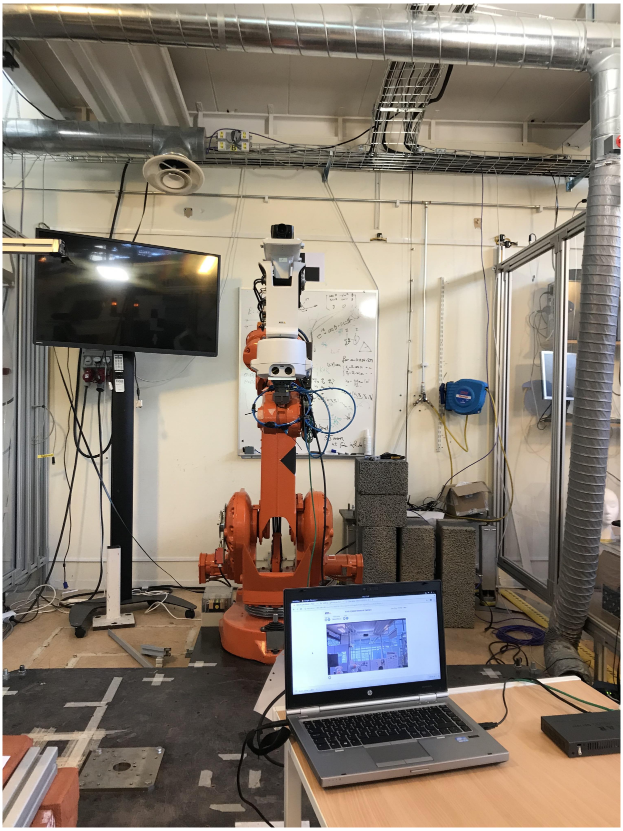


Figure 9.5: Experiment setup in the Robotlab at LTH.

9.3 Gränsö experiments

Some real simulations of the stabilized unit in the envisaged environments were conducted in Gränsö, Sweden. The positioning unit was mounted on a small boat with an outboard motor, see Figs. 9.6 and 9.7, which was driven at different velocities in altering directions. Note that the final mounting of the positioning unit on the boat was with the camera housing facing forward, as in 9.7. During these experiments, the EIS was turned on, and run in parallel with the stabilization algorithm, to test if there would be any impacts in the functionality regarding this. The visit to Gränsö was made possible thanks to a collaboration with WASP [WASP, n.d.], mentioned earlier, and Axis. The results from the experiments were used as material for discussion and basis for the final product.

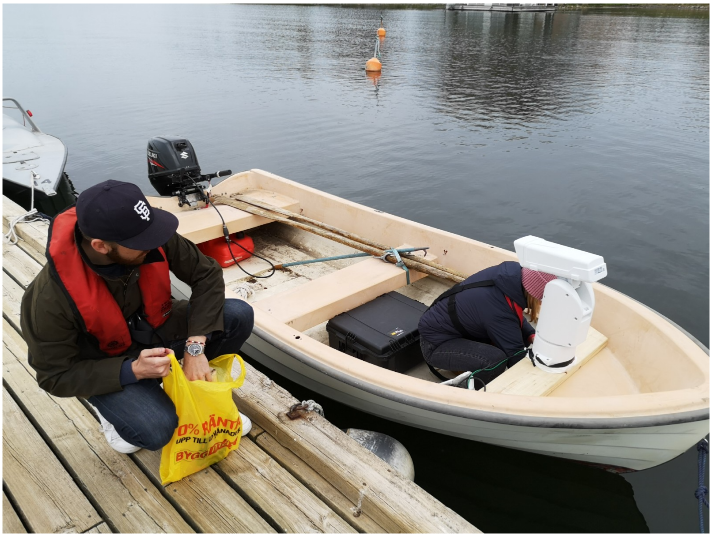


Figure 9.6: Mounting of the positioning unit on the boat.

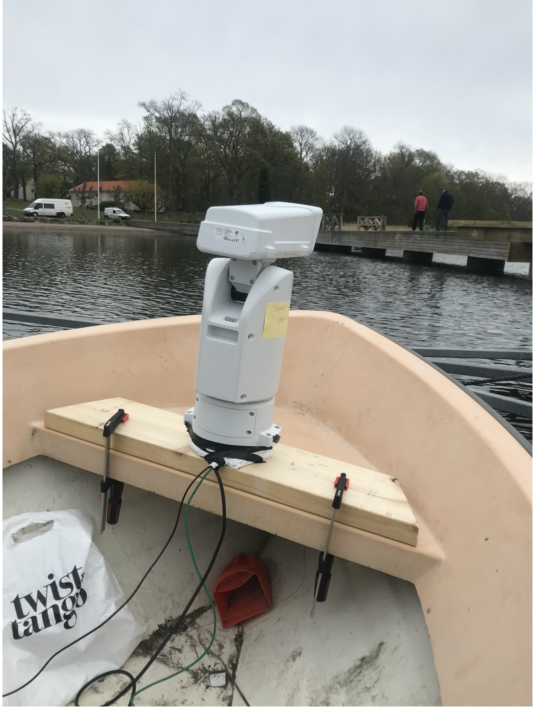


Figure 9.7: Mounting of the positioning unit on the boat, in detail.

10

Results

This chapter presents the results obtained from the aforementioned inhouse experiments as well as the experiments carried out in Gränsö and Robotlab at LTH.

10.1 Results from functionality tests

Fig. 10.1 shows the result from the reference creation, with the reference signals in roll, pitch (tilt) and yaw (pan) (red lines) as well as their corresponding measured angles from the IMU data acquired from the in-house experiments (blue lines).

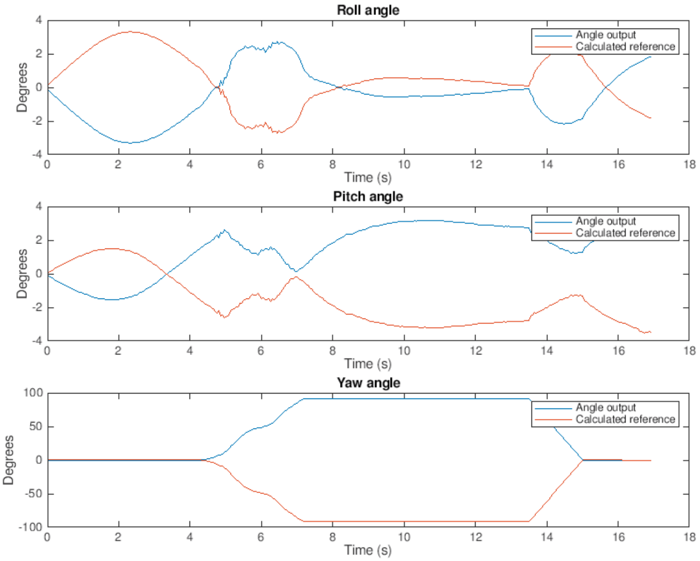


Figure 10.1: Calculated reference and angle measurements from IMU data.

10.2 Robotlab results

Figs. 10.2 to 10.5 show the results from the experiments carried out in Robotlab. All plots visualize the performance of the different PID controllers, by showing the input reference signal plotted against the output signal, which is the encoder data. The first two mentioned figures show the reference signal and the encoder values for the pan motor and a zoomed-in frame of the results, respectively. The latter two show the reference signal and the encoder values for the tilt motor, again with a zoomed-in frame of the results.

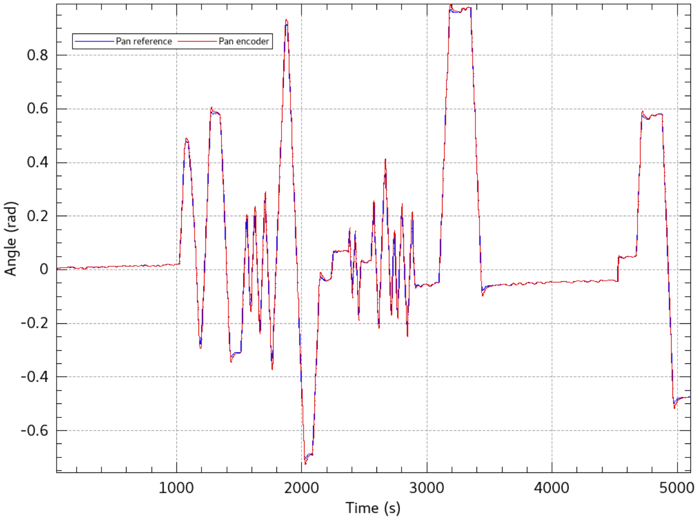


Figure 10.2: Pan reference tracking from ABB robot experiments.

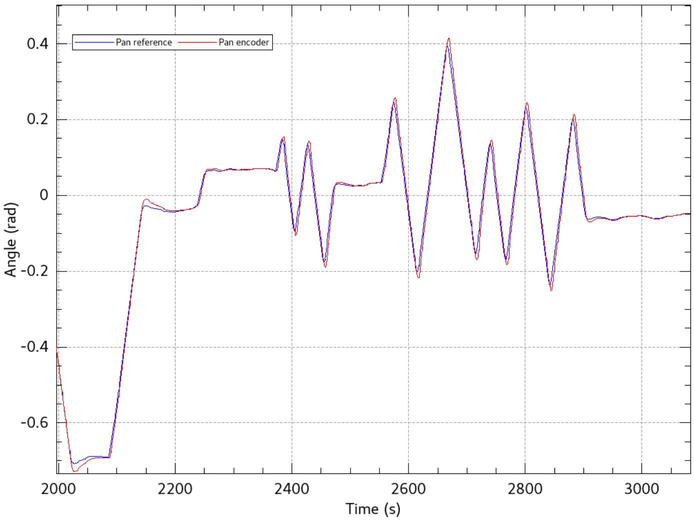


Figure 10.3: Pan reference tracking from ABB robot experiments, zoomed-in.

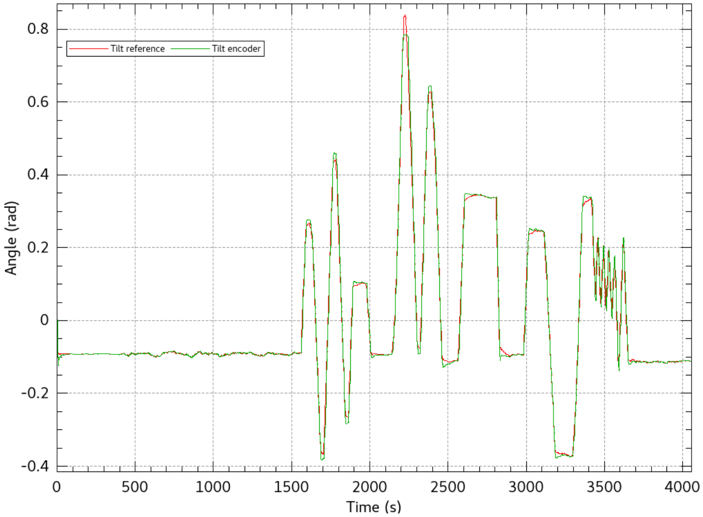


Figure 10.4: Tilt reference tracking from ABB robot experiments.

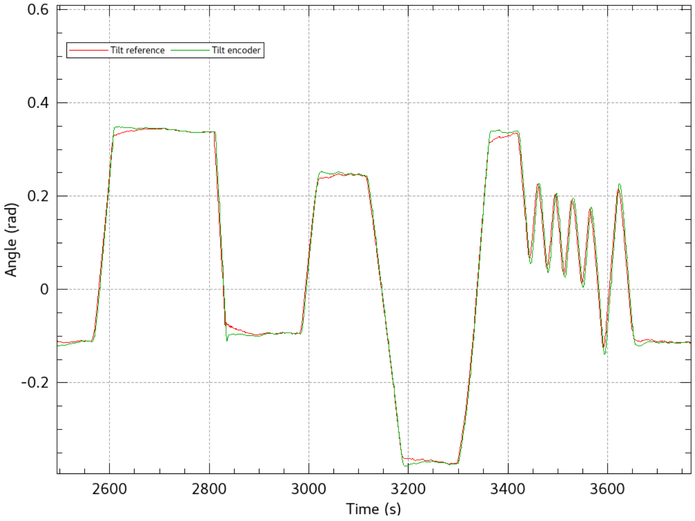


Figure 10.5: Tilt reference tracking from ABB robot experiments, zoomed-in.

In order to study the behaviour of the tilt motor at higher frequencies, another experiment was conducted in Robotlab. During this experiment, the fifth joint of the ABB robot, i.e. the joint with the DOF along the pitch axis which is closely located to the positioning unit, was jogged at a high frequency. The results can be seen in Figs. 10.6 and 10.7 where the latter one is a zoomed-in frame of the results in the first figure. In these figures, the reference signal, encoder values and the output signal were plotted and studied.

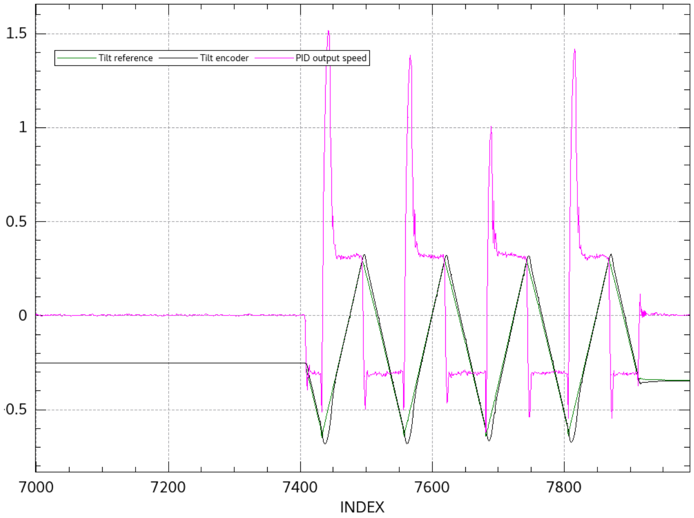


Figure 10.6: Tilt overshoot

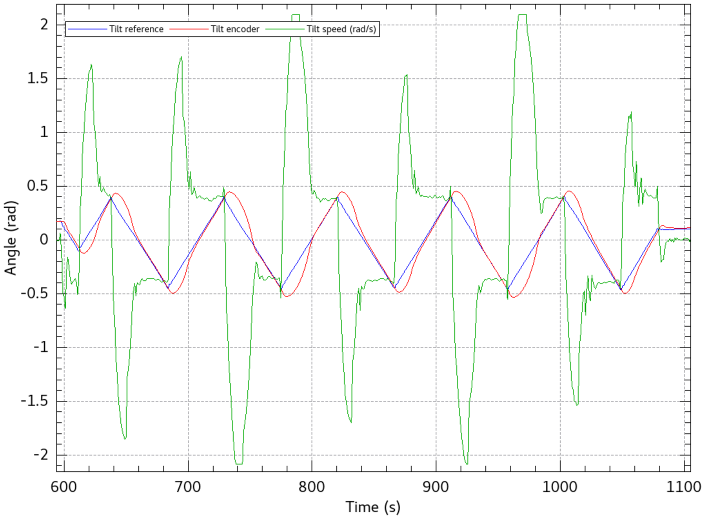


Figure 10.7: Tilt overshoot, zoomed-in.

Even though the positioning unit is not equipped with a motor along the roll axis, an experiment where only the roll angle of the ABB robot was altered was conducted. This was done to see how the pan- and tilt motors would behave in such a situation. Fig. 10.8 shows the results from this experiment, where the reference signal for roll, pitch and yaw as well as their corresponding measured angles were plotted.

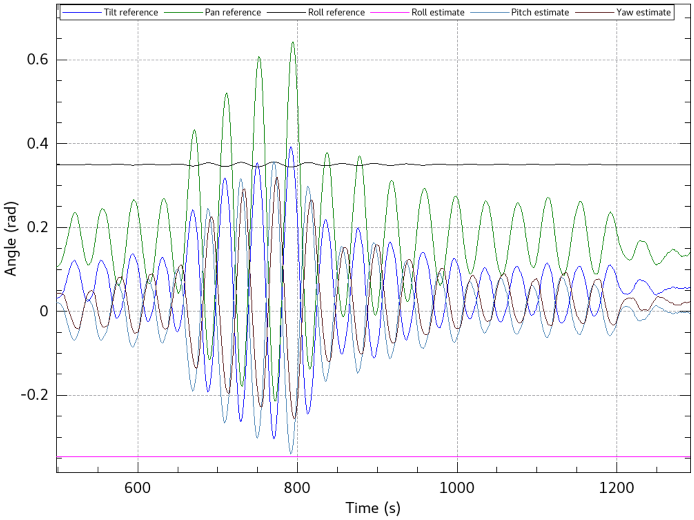


Figure 10.8: Oscillating behaviour at larger roll angles.

10.3 Results from Gränsö/Boat experiments

Just like during the Robotlab experiments, the Gränsö experiments enabled evaluating the performance of the different PID controllers by plotting the same signals against each other. Figs. 10.9 and 10.10 show the PID reference following for the tilt motor and a zoomed-in frame of the results, respectively, where the reference signal and the actual tilt encoder values were plotted. Fig. 10.11 shows the error between the tilt reference signal and the tilt encoder values.

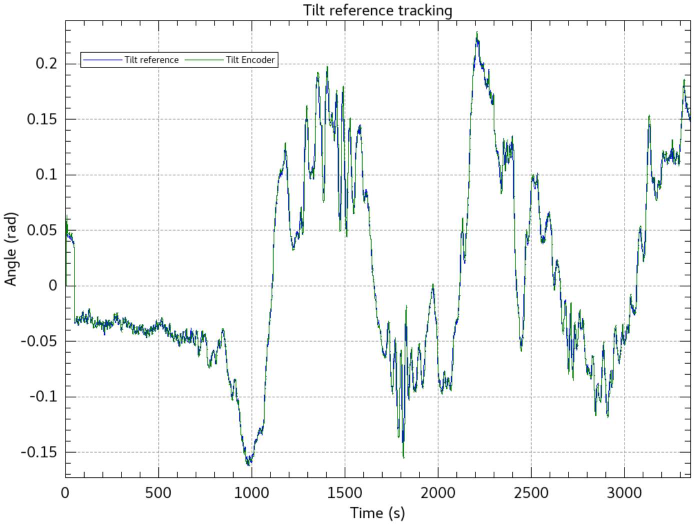


Figure 10.9: Tilt reference tracking from boat experiment.

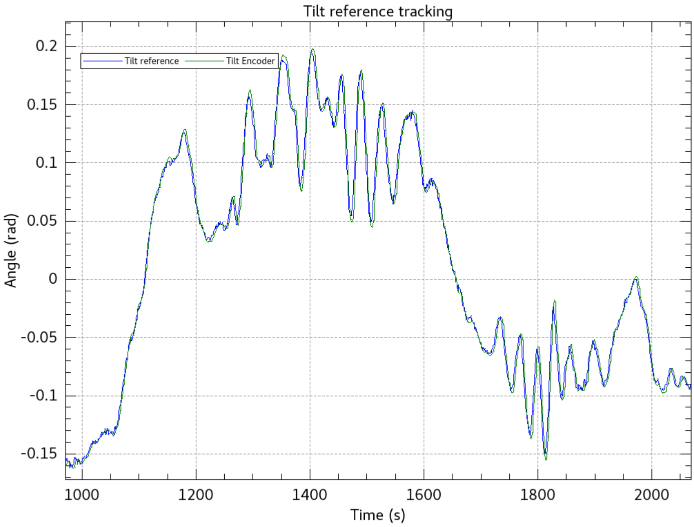


Figure 10.10: Tilt reference tracking from boat experiment, zoomed-in.

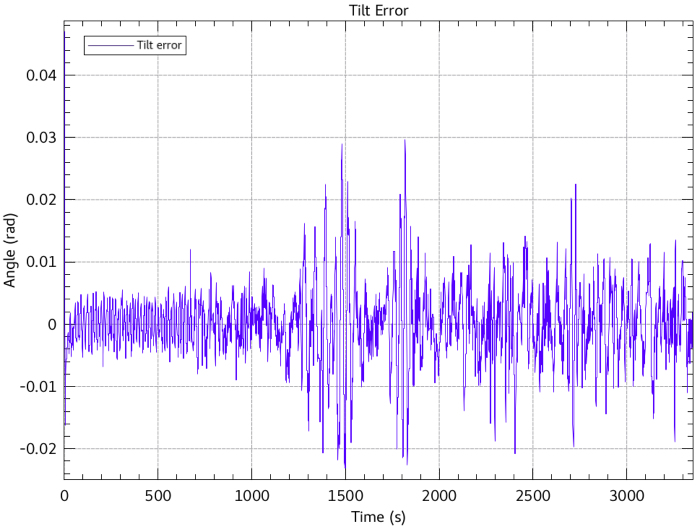


Figure 10.11: Tilt error

Figs. 10.12 and 10.13 show the reference tracking for the pan motor and a zoomed-in frame of the results, respectively. Like the aforementioned figures, the reference signal and the pan encoder values were plotted. In Fig. 10.12, there is a large spike at approx. 2200 s which is caused by the fact that the encoder values do not range from 0° to 360° but rather from $-180°$ to 180°. The same goes for the reference signal and will cause the values to jump from $-180°$ to 180°, or $-\pi$ radians to π radians as in the figure. Fig 10.14 shows the error between the pan reference signal and the pan encoder values.

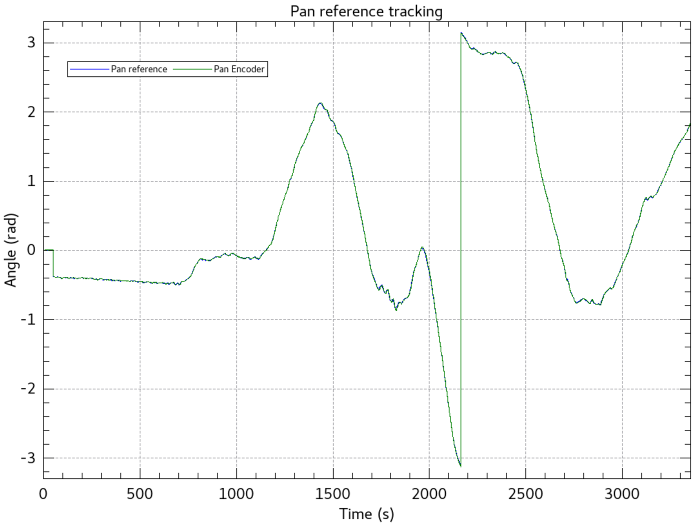


Figure 10.12: Pan reference tracking from boat experiment.

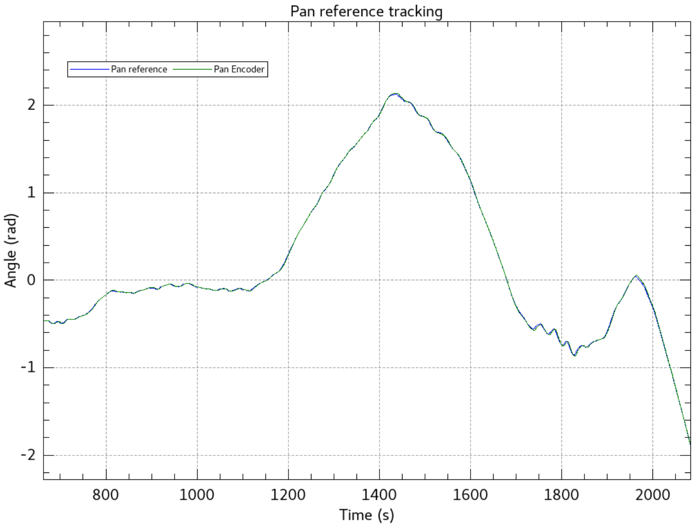


Figure 10.13: Pan reference tracking from boat experiment, zoomed-in.

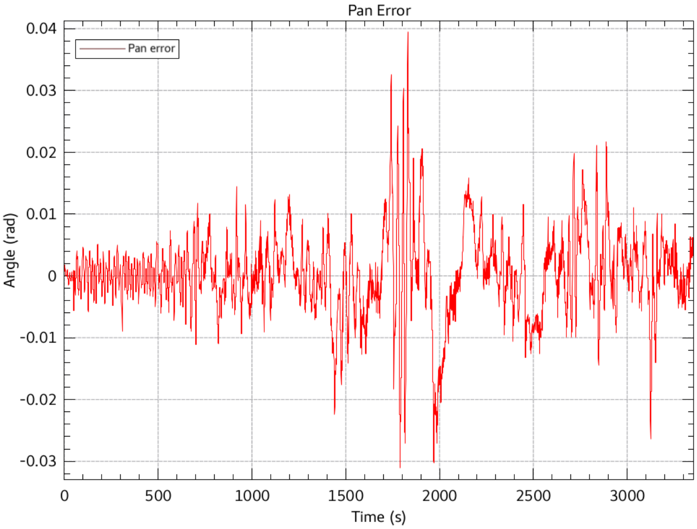


Figure 10.14: Pan error

11

Discussion

This chapter includes the discussion of the results presented in the previous chapter as well as more general discussions and conclusions regarding the thesis as a whole and ideas for further work.

11.1 Results

The results from the functionality test regarding the reference signal calculation shown in Fig. 10.1 showed desired results. The plot clearly shows that the reference signals match the measured angles in size but with opposite sign, meaning that the created reference signals will counteract the angular disturbances.

The reference tracking tests that were carried out in Robotlab all showed that the actual encoder values followed the reference signal apart from some minor overshoots at larger angles. However, the encoder values for the pan motor does follow the reference signal slightly better with smaller overshoots than the tilt motor. This behaviour indicates that some further investigation of the respective dynamics of the motors needs to be considered. An assumption was made at an early stage that the two motors work somewhat similarly. At this stage, it is clear that that was not the case.

There are other factors that can be the cause of the different behaviours in pan and tilt. Among these are the fact that they are affected differently by the gravitation and that they do not have the same mechanical prerequisites.

In order to get a better understanding of the overshoots present in the results, the results in Figs. 10.6 and 10.7 were studied. The overshoot can be seen right after the joint of the robot changed direction, causing the output signal (speed) to increase in size. This behaviour did, however, only occur when the joints of the robot changed direction quickly, which is how the dynamics of the robot is designed. When the same joint was jogged with smoother changes in direction, the overshoot decreased. Since the S-curves handling the motion controllers are designed to peak in acceleration at the start in a change of velocity, this is hard to counteract for, since it would require a reconfigure of the motion controllers. Thus, it can be concluded that the tilt stabilization cannot handle disturbances with velocities above approx. 90°/s (number taken from Fig. 10.6). Anyhow, from the experiments carried out in a real scenario, this speed was not reached by far, why it was concluded that this would not affect the quality of the results in a negative way.

It is more difficult to draw a conclusion of the limits of the pan stabilization, since the control signal follows the reference well. While performing the experiments in Robotlab, some extremes were tested to see if the pan stabilization could not counteract for certain harsh movements. The only thing found was that some overshoots were introduced when the velocity was approx. 120°/s. Since this was concluded to be unlikely to happen in a real scenario, further tests were disregarded. 120°/s is also the velocity limit of the motors, why further tests in even greater velocities could not be tested.

The results in Fig. 10.8, where only the roll angle was increased, clearly shows that both the pitch- and yaw angle estimates as well as the corresponding reference signals start to oscillate whenever the roll angle grows sufficiently large. This has most likely to do with the fact that the center of rotation of the positioning unit is located with an offset from the pan- and tilt motors and the IMU. Therefore, the motors must compensate much more for the roll disturbances. If the motors and the IMU had been located in the center of rotation, there would not be any need for this type of compensation. Instead, the picture would only rotate along its roll axis which still could not be fixed due to the lack of a roll motor. This behaviour occurred at roll angles larger than approx. 20° which, according to the gathered data from Karlskrona, is less likely to happen in a real life scenario. Therefore, the solution to this problem was simply to saturate the roll angle just below 20° which gave preferable results.

During the Gränsö experiments, the error, in both pan and tilt, between the reference signal and the encoder values were plotted and presented in Figs. 10.11 and 10.14. As can be seen in both figures, the error is close to zero during the whole experiment. There is no underlying oscillations present in neither tilt nor pan which can occur if the control algorithm is malfunctioning. Instead, the error behaves like so-called white noise which is expected.

The conclusion drawn from the real life experiments is mainly that the disturbances are significantly smaller than the ones introduced in the Robotlab experiments. The change of direction, especially in tilt, were also much smoother which resulted in smaller overshoots in the reference tracking.

11.2 Sources of error

In addition to the sources of error that could be seen in the results, there are a few others that, to some extent, affected the reliability of the stabilization algorithm. One of these are the placement of the IMU. Since it is located in the body of the positioning unit, it will rotate together with the pan motor. This can cause some trouble when running the algorithm since one has to take this into consideration when processing the gyroscope measurements. The algorithm in this thesis works in a way that the pan encoder values are subtracted from the yaw angle obtained from the integrated gyroscope measurements. Even though this solution gave desired results, its could cause some misleading measurements or even cause the gyroscope to drift even more.

Another source of error regarding the IMU is the slight placement offset of approx. 2.75° as described in Section 8.2. The solution to eliminate this offset in the start up phase of the algorithm proved to be suitable based on the conditions but the ideal solution would be to install the IMU more precise.

The acceleration limits for the two motors were set to $200^\circ/s^2$. These limits could have been altered for especially the tilt motor in order to investigate possible solutions to the different behaviour between the motors. Increasing the limit could have improved the performance of the tilt motor's stabilization but would have resulted in the need to design a new controller for the motor. However, since the results in both Robotlab and Gränsö were considered desirable and due to the lack of time, this was not investigated further.

12

Conclusions

Overall, the authors are satisfied with the results. The planning and structure of the thesis were followed and considered reasonable. No part of the thesis was considered rushed or too time consuming. Since both authors were equally involved in the different parts of the thesis, both feel that they have developed new knowledge and experience in regards to the given topic.

A general conclusion that could be drawn from the results is that the overall goal of the gyro stabilization is fulfilled, since the stabilization algorithm works for both the pan- and tilt axis. The other goals in this thesis are also considered fulfilled; data was collected from several scenarios, different motion detecting units (IMU:s) were discussed and scenarios for gyro stabilization were investigated and the stabilization algorithm tested in one of these scenarios, and proved to work. Even though no tests were made in the other scenario investigated, terrain, it was concluded that the resulting implementation would have proven to work successfully in this scenario as well, considering the previous gyroscope data shown from this environment.

Although the results were satisfying to prove the concept of this thesis, there is always room for improvement. For example, alternate placements of the chosen IMU could have been more thoroughly tested in order to investigate if the performance could have been enhanced when the IMU is not placed in a position where it rotates together with the pan movements. Since the pan- and tilt behaviours differ, it would have been interesting to further investigate options for modeling of the motors and ways to perform the system identification, in order to eliminate this source of error. For example, further investigation of the S curves in the system and integration of this in the system identification would have been interesting to conduct. Also, other controllers could have been investigated for the application. Only PI- and PID controllers were investigated for the application, since there already existed a ready-to-use PID controller in the software, and the fact that the PID controller was proven to work for the given system. Although, it could have been interesting to try other options in terms of control, to further enhance the performance of the algorithm. For example, an MPC controller or an MIMO controller could have been investigated for this purpose.

12.1 Further work

To further enhance the performance of the stabilization algorithm, some improvement areas have been identified, although outside the scope of this thesis. One improvement that could enhance the stabilization algorithm would be to enable compensating for disturbances in the roll angle. Since there is not a possibility to do this using the current motors in the positioning unit, it would have to be done in the current EIS of the network camera. This would give an enhanced stabilization since counteracting for disturbances would be possible about all rotation axes. Another possible improvement would be to enable steering of the joystick during the stabilization algorithm. As of now, the algorithm takes the current position of the positioning unit as the desired position and holds this position throughout the stabilization algorithm. The operator is therefore unable to change this position during the algorithm. If this functionality is desired by the operator, it could be interesting to investigate how to implement so that the desired position could be altered during the stabilization algorithm. It would also have been interesting to investigate if it would have been possible to implement some kind of mid-ranging between the EIS and the stabilization algorithm. In the beginning of this thesis work, it was briefly discussed but since these systems are implemented on two separate platforms, no easy way to combine them communication wise was found, and therefore the idea was disregarded. Since it it possible to run the two systems in parallel though, it could be interesting to look more into the possibility to fuse them together in some way. Furthermore, it would be interesting to look more into how to compensate for or eliminate the gyroscopic drift, since this drift is unwanted.

Bibliography

ABB Global (2019). *IRB 2400.* URL: https://new.abb.com/products/robotics/industrial-robots/irb-2400 (visited on 2019-05-06).

Årzén, K.-E. (2019). *FRTN01 - Real Time Systems - Lecture Notes.* LTH, Automatic Control.

Åström, K. J. and R. M. Murray (2012a). *PID Control.* URL: http://www.cds.caltech.edu/~murray/amwiki/index.php?title=PID_Control (visited on 2019-03-05).

Åström, K. J. and R. M. Murray (2012b). *Transfer Functions.* URL: http://www.cds.caltech.edu/~murray/amwiki/index.php?title=Transfer_Functions (visited on 2019-03-05).

Åström, K. J. and R. M. Murray (2018). *Feedback Systems - An Introduction for Scientists and Engineers.* PRINCETON UNIVERSITY PRESS.

Axis (2019). *Axis Communications AB.* URL: https://www.axis.com/sv-se (visited on 2019-01-29).

Basic Air Data (2011). *Inertial Measurement Unit Placement.* URL: https://www.basicairdata.eu/knowledge-center/compensation/inertial-measurement-unit-placement/ (visited on 2019-02-12).

Beavers, I. (2017). *The Case of the Misguided Gyro.* URL: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-139.html# (visited on 2019-03-19).

Bergdahl, L. (2009). "Wave-Induced Loads and Ship Motions". *Department of Civil and Environmental Engineering, Division of Water Environment Technology, Chalmers University of Technology, Goteborg.*

CHRobotics (n.d.). *Understanding Euler Angles.* URL: http://www.chrobotics.com/library/understanding-euler-angles (visited on 2019-02-05).

Day, M. (n.d.). *Extracting Euler Angles from a Rotation Matrix.* Insomniac Games.

Drescher, A. and J. Hambleton (2010). "Geotechnics and Terramechanics". *University of Minnesota: 58th Annual Geotechnical Engineering Conference.*

Freidovich, L. B. (2017). *Control Methods for Robotic Applications*. Umeå Universitet.

Gustavi, M. and L. Andersson (2017). *Implementation of Control Algorithm for Mechanical Image Stabilization*. Master's thesis. Lund University.

Hua, J. and M. Palmquist (1995). "A Description of SMS - A Computer Code for Ship Motion Simulation". *Naval Architecture, Department of Vehicle Engineering, KTH, Stockholm.*

InvenSense (2014). *Accelerometer and Gyroscope Design Guidelines*. URL: `https://store.invensense.com/datasheets/invensense/Accelerometer_Gyroscope_Design_Guidelines.pdf` (visited on 2019-02-12).

ISTVS, International Society for Terrain.-Vehicle Systems. (2019). *ISTVS Themes*. URL: `https://www.istvs.org/` (visited on 2019-01-29).

Kopetz, D. H. (2002). *Sensor Fusion in Time-Triggered Systems*. Doctoral Dissertation. Technischen Universität Wien.

Lewin, C. (n.d.). "Mathematics of Motion Control Profiles". *Performance Motion Devices - Motion control at it's core.*

Ljung, L. (2007). "System Identification - Theory For the User". *Department of Automatic Control, Linköping University.*

Ljung, L. and T. Glad (2000). *Control Theory: Multivariable and Nonlinear Methods*. CRC Press.

Ljung, L. and T. Glad (2003). *Modellbygge och simulering*. Studentlitteratur. ISBN: 9789144024431.

Mathworks (2019). *Atan2*. URL: `https://se.mathworks.com/help/matlab/ref/atan2.html` (visited on 2019-02-12).

Ovegård, E. (2009). "Numerical simulation of parametric rolling in waves". *Naval Architecture, Department of Vehicle Engineering, KTH, Stockholm.*

Perez, T. (2005). *Ship Motion Control*. Springer-Verlag London.

Plantenberg, K. and R. Hill (2013). *Pure rotation - gears and belts*. URL: `http://www.sdcpublications.com/multimedia/978-1-58503-767-4/files/krb/krb_pr_page3.htm` (visited on 2019-03-28).

Salmony, P. (n.d.). *IMU Attitude Estimation*. URL: `http://philsal.co.uk/projects/imu-attitude-estimation` (visited on 2019-03-06).

Saputra, H. M., Z. Abidin, and E. Rijanto (2013). "IMU Application in Measurement of Vehicle Position and Orientation for Controlling a Pan-Tilt Mechanism". *Mechatronics, Electrical Power, and Vehicular Technology* 4, pp. 41–50.

ScienceDirect (2019). *Magnetometer*. URL: `https://www.sciencedirect.com/topics/engineering/magnetometer` (visited on 2019-02-04).

SEGGER (n.d.). *J-Link Debug Probes*. URL: https://www.segger.com/products/debug-probes/j-link/ (visited on 2019-05-17).

Sen, P. C. (1997). *Principles of Electric Machines and Power Electronics*. John Wiley & Sons.

SparkFun Electronics (n.d.). *Accelerometer, Gyro and IMU Buying Guide*. URL: https://www.sparkfun.com/pages/accel_gyro_guide (visited on 2019-02-11).

SparkFun Electronics (2013a). *Accelerometer Basics*. URL: https://learn.sparkfun.com/tutorials/accelerometer-basics/all (visited on 2019-02-04).

SparkFun Electronics (2013b). *Gyroscope*. URL: https://learn.sparkfun.com/tutorials/gyroscope/all (visited on 2019-02-04).

Tala, D. K. (2014). *Digital Combinational Logic Part-iii*. URL: http://www.asic-world.com/digital/combo3.html (visited on 2019-03-20).

TDK InvenSense (2019a). *MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices*. URL: https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/ (visited on 2019-02-01).

TDK InvenSense (2019b). *MPU-9250 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device*. URL: https://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/ (visited on 2019-02-01).

Texas Instruments Inc. (1997). “DSP Solutions for BLDC Motors”.

ToolBox, E. (2008). *Belts - Power Transmission and Efficiency*. URL: https://www.engineeringtoolbox.com/belt-transmission-power-efficciency-d_1378.html (visited on 2019-03-28).

Vectornav (n.d.). *Magnetometer*. URL: https://www.vectornav.com/support/library/magnetometer (visited on 2019-02-04).

WASP (n.d.). *Wallenberg ASP*. URL: https://wasp-sweden.org/ (visited on 2019-04-17).

Yedamale, P. (2003). “Brushless DC (BLDC) Motor Fundamentals”. *Microchip Technology Inc.*

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER’S THESIS

Date of issue
June 2019

Document Number
TFRT-6082

Author(s)
Wilhelm Andrén
Ella Hjertberg

Supervisor
Torkel Niklasson, Axis Communications AB
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden
Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Gyro Stabilization of a Positioning Unit

Abstract
Mounting cameras on motorized objects has become possible in greater extent due to the emerging camera technology during the past few decades. This application could prove useful in several areas such as search and rescue operations, surveillance or even news monitoring. One hardship that this brings is the difficulty of keeping the camera unit stable while its setting is not. In the situations mentioned above, it would be desired to keep the camera unit in the same attitude during the operation, regardless of the impacts from the surrounding environment. The goal of this thesis has been to solve this problem using gyro stabilization by implementing a stabilization algorithm to be added in the camera unit’s software.

The idea behind gyro stabilization is to counteract for external disturbances causing the camera unit to dislocate from its desired attitude. This is realized by controlling two BLDC, Brushless DC, motors in a positioning unit (PU) provided by Axis Communications AB. The PU could be rotated about two axes (using the two motors) perpendicular to each other, pan and tilt, and could hold either a network video camera, or a set of illuminators. The control of the motors is achieved by utilizing data from an IMU, Inertial Measurement Unit, including a gyroscope and an accelerometer, whose measurements are fused in a Kalman filter. This data, together with an estimation of the orientation of the PU, is used to calculate the error between the actual position and the desired position of the PU. The orientation estimation is found by relating the coordinate system of the PU to a fixed, global coordinate system. The calculated error is then minimized by using a PID controller to control the velocity of the pan- and tilt motors and ultimately the position of the PU.

Most of the calculations and simulations regarding the stabilization algorithm were carried out in MATLAB Simulink, and later implemented in the PU’s software, written in C code. The final algorithm was tested both on an ABB IRB2400 robot, and mounted on a small boat driven by an outboard motor, made possible by an collaboration with the research program WASP. The results were satisfying and the PU was successfully stabilized, which was the main goal of the thesis. However, there is room for some minor improvements, for example improving the modelling of the motors.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-97

Recipient’s notes

Security classification

http://www.control.lth.se/publications/