

Implementation of control algorithm for mechanical image stabilization

Magnus Gustavi

Louis Andersson



LUND
UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6042
ISSN 0280-5316

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

© 2017 by Magnus Gustavi & Louis Andersson. All rights reserved.
Printed in Sweden by Tryckeriet i E-huset
Lund 2017

Abstract

Cameras mounted on boats and in other similar environments can be hard to use if waves and wind cause unwanted motions of the camera which disturbs the desired image. However, this is a problem that can be fixed by applying mechanical image stabilization which is the goal of this thesis.

The mechanical image stabilization is achieved by controlling two stepper motors in a pan-tilt-zoom (PTZ) camera provided by Axis Communications. Pan and tilt indicates that the camera can be rotated around two axes that are perpendicular to one another.

The thesis begins with the problem of orientation estimation, i.e. finding out how the camera is oriented with respect to e.g., a fixed coordinate system. Sensor fusion is used for fusing accelerometer and gyroscope data to get a better estimate. Both the Kalman and Complementary filters are investigated and compared for this purpose. However, the Kalman filter is the one that is used in the final implementation, due to its better performance.

In order to hold a desired camera orientation a compensation generator is used, in this thesis called reference generator. The name comes from the fact that it provides reference signals for the pan and tilt motors in order to compensate for external disturbances. The generator gets information from both pan and tilt encoders and the Kalman filter. The encoders provide camera position relative to the camera's own chassi. If the compensation signals, also seen as reference values to the inner pan-tilt control, are tracked by the pan and tilt motors, disturbances are suppressed.

In the control design a model obtained from system identification is used. The design and control simulations were carried out in the MATLAB extensions Control System Designer and Simulink. The choice of controller fell on the PID.

The final part of the thesis describes the result from experiments that were car-

ried out with the real process, i.e. the camera mounted in different setups, including a robotic arm simulating sea conditions. The result shows that the pan motor manages to track reference signals up to the required frequency of 1Hz. However, the tilt motor only manages to track 0.5Hz and is thereby below the required frequency. The result, however, proves that the concept of the thesis is possible.

Acknowledgements

First we would like to thank Axis Communications and especially Björn Ardö, Johan Nyström and Anders Sandahl for their assistance during the project. Furthermore, we would like to thank Stig Frohlund for producing an adapter plate that was used for mounting the camera on a robotic arm in the test phase of the thesis.

We would also like to thank our advisor, professor Anders Robertsson, at the Department of Automatic Control, LTH, for his help and feedback to ideas during the whole thesis.

Abbreviations and symbols

MCPU	Main Central Processing Unit
MCU	Microcontroller Unit
HMI	Human Machine Interface
IMU	Inertial Measurement Unit
OIS	Optical Image Stabilization
EIS	Electronic Image Stabilization
PoE	Power over Ethernet
NED	North-East-Down
PID	Proportional-Integral-Derivative
SISO	Single-Input-Single-Output
IMC	Internal Model Control
PTZ	Pan-Tilt-Zoom
h	Sampling time
I ² C	Inter-Integrated Circuit
LTH	Lunds tekniska högskola
MPC	Model Predictive Control
DC	Direct current
ψ, θ, ϕ	Euler angles relative NED
ψ_c, θ_c, ϕ_c	Euler angles relative camera chassi
ψ_d, θ_d, ϕ_d	Desired Euler angles relative NED

Contents

1. Introduction	11
1.1 Background	11
1.2 Goals and problem formulation	12
1.3 Delimitations	12
1.4 Individual contributions	13
1.5 Outline of the thesis	13
2. Image stabilization	14
2.1 Introduction	14
2.2 Optical image stabilization	15
2.3 Sensor shift image stabilization	16
2.4 Electronic image stabilization	16
2.5 Mechanic-based image stabilization	16
3. Control and system overview	17
3.1 Platform	17
3.2 Actuators, motor drivers & motion controllers	18
3.3 Sensors	18
3.4 System architecture	19
3.5 Control concept	20
3.6 Disturbances at sea	20
4. Orientation and sensor fusion	22
4.1 Coordinate systems	22
4.2 Attitude estimation with accelerometers	25
4.3 Sensor fusion	26
4.4 Result and discussion	31
4.5 Conclusion	34
5. Reference generation	36
5.1 Extracting Euler angles from a rotation matrix	37
5.2 Result and discussion	38

6. Modelling	41
6.1 Physical modelling	42
6.2 System identification	46
7. Trajectory generation	49
7.1 Motion profiles	50
7.2 Motion controllers	53
7.3 Discussion	55
8. Control and implementation	56
8.1 PID-control	57
8.2 Control structures	60
8.3 Control design	62
8.4 Hardware implementation	69
9. Results	74
9.1 Closed loop step response	74
9.2 Closed loop sinusoidal response	76
9.3 Shake experiments	78
10. Discussion and conclusions	84
10.1 Orientation and sensor fusion	84
10.2 Reference generation	84
10.3 Modelling	85
10.4 Control and implementation	85
10.5 Future work	86
10.6 Conclusions	86
Bibliography	87
A. Velocity trajectory generation	90
A.1 Velocity mode	91
A.2 Cases	92

1

Introduction

Cameras mounted on vehicles are nowadays more common than before, mostly because of the development in the area of autonomous systems. The role of the camera can however be different but in many cases it is desired to keep the camera image stable or even, if it is possible, to track certain objects. Moreover it must not only concern autonomous vehicles but also conventional vehicles. In this thesis image stabilization on boats has been of particular interest.

1.1 Background

Axis Communications [Axis, 2017] is a company that today is using cameras with electronic image stabilization. However, it can only compensate for disturbances with low amplitude as the electronic stabilization reduces the size of the image shown. Due to this the stabilization could be improved by using mechanical image stabilization as well, namely, control of the camera motors to compensate for external disturbances. In this way it would be possible to suppress larger disturbances.

As mentioned above, the area of use that has been considered in this thesis is stabilization onboard boats, where waves and wind affect the position of the camera in a more dramatic way compared to land-based mountings. The disturbance frequency is however smaller compared to many other environments, thereby making it especially suitable for mechanical stabilization which cannot operate that fast.

To measure the influence caused by disturbances, a set of sensors have been used. All the sensors were already integrated in the camera shell and software for retrieving data was also available.

The camera construction is similar to that of a gimbal, meaning that it can move almost freely around two orthogonal axes (pan and tilt). This kind of setup is also often used for handheld cameras where image stability is of importance. In Figure 1.1 the construction can be seen. Pan and tilt correspond to rotations around the k -axis and e -axis, respectively.

1.2 Goals and problem formulation

The final goal of this thesis is to have a camera that manages to stabilize the camera image when it is exposed to disturbances between 0 and 1Hz and with respect to an amplitude that is too large for the electronic image stabilization to handle. To reach this main goal a set of problem formulations were set up:

- Orientation estimation: how can the camera orientation be measured?
- Reference generation: how can the pan and tilt motors be given commands that keeps the camera in the right orientation?
- Control: how can the pan and tilt motors be controlled in order to follow the command/reference signals?

1.3 Delimitations

In this thesis there has been no focus on optimizing code in the implementation that was done. The main focus was instead to prove the concept and to obtain a working prototype.

Regarding control only one controller type was investigated, meaning that the controller implementation might not be optimal.

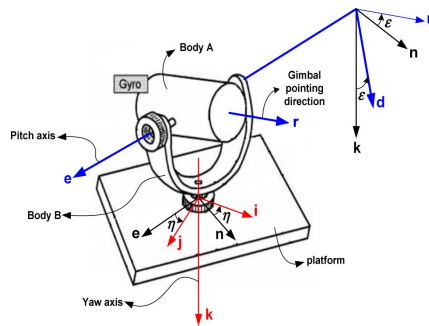


Figure 1.1 Two-axis gimbal system [Abdo et al., 2013]

1.4 Individual contributions

In order to make the work of the project more efficient it was divided between the participants. Magnus focused on the orientation and sensor fusion, the reference generation and the mechanical modelling, whereas Louis focused on the system identification, the stepper motor modelling and the trajectory generation. The control design and the hardware implementation were done together, however, Louis did a greater part in the hardware implementation and Magnus worked more with the control design.

1.5 Outline of the thesis

In Chapter 2 image stabilization is described in general in order to give an introduction to the subject. Moreover, different techniques for stabilization are presented, including mechanical image stabilization.

Chapter 3 gives an overall view of the system that has been used. Furthermore a description of the control concept and possible disturbances are presented.

Chapter 4 describes the solution to the problem of finding camera orientation. It begins with a presentation of coordinate systems and Euler angles. It also goes into the area of sensor fusion and in the end experimental results are presented.

Chapter 5 presents reference generation which describes how compensation signals for disturbance suppression are produced. The compensation signals can also be seen as reference signals since the pan and tilt motors are supposed to track them.

Chapter 6 and 7 present the modelling done in the project. Three different approaches are presented, namely: Physical modelling, system identification and motion profiles.

Chapter 8 begins with a general description of control theory. Furthermore, the control design for the pan and tilt motors is described and simulation results are presented. In the end of the chapter hardware implementation for all project parts are also described.

Chapter 9 presents experimental results regarding the pan and tilt motor control. The results come from three different experiments.

In Chapter 10 a discussion and conclusion regarding the project as a whole is presented. Moreover, possible future work is described.

2

Image stabilization

2.1 Introduction

Image stabilization is a family of techniques used to reduce blurring associated with the vibration on a camera or other image devices. These techniques will not prevent motion blur caused by movement of the target subject but only rotational movement of the image device itself. However, there also exist six-axes solutions where also linear movement can be compensated for, but this is not considered here. Three different rotations can be compensated for depending on the technique used, see Figure 2.1. Some of these techniques require vibration sensing, therefore sensors such as gyroscopes and accelerometers are needed.

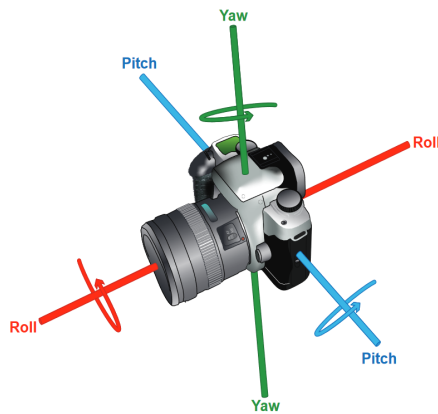


Figure 2.1 Axes of motion [ROHM-Semiconductor, 2017]

2.2 Optical image stabilization

Optical Image Stabilization, or OIS in short, is a mechanical technique that uses the optic part of a camera. The two main methods of OIS in compact camera modules are implemented by either moving the position of the lens (lens shift) or the module itself (module tilt) [ROHM-Semiconductor, 2017]. For bigger camera models lens shift is used [Golik, 2006]. Vibration on the camera module can cause misalignment of the optical path between the focusing lens and the center of the image sensor. In a lens shift method only the lens within the camera module is controlled and used to realign the optical path to the center of the image sensor. In a module tilt method the movement of the entire module is controlled, including the fixed lens and the image sensor. This method allows for a greater range of movement compensation, but the trade off is increased module height. Compared to other image stabilization methods such as EIS, see more in Section 2.4, OIS systems reduce image blurring without significantly sacrificing image quality, especially for low-light and long-range image capture. However, due to the addition of actuators and the need for power driving sources the OIS modules tend to be larger and as a result expensive to implement. EIS just need some sensors or no additional hardware depending on what type of algorithm that is used. OIS can not compensate for camera roll because of the symmetrical properties of the lens. Even if the lens is rotated its orientation remains the same relative to the image sensor [ROHM-Semiconductor, 2017].

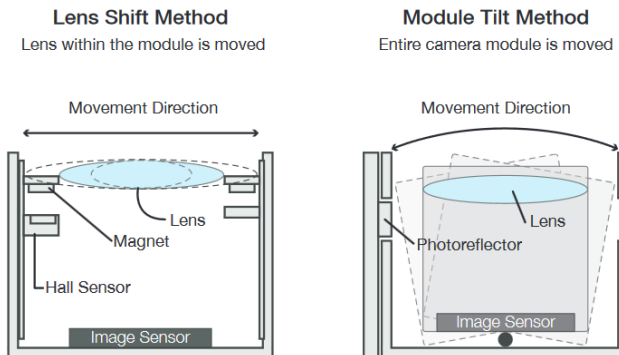


Figure 2.2 Main Methods of OIS Compensation [ROHM-Semiconductor, 2017]

2.3 Sensor shift image stabilization

Sensor shift image stabilization system works by moving the camera's sensor around the image plane using actuators. With help of accelerometers and gyroscopes it can then sense the vibration that occurs on the camera module. The advantage with moving the image sensor, instead of the lens as in optical image stabilization, is that the stabilization can be done regardless of the lens used. Another advantage is that some sensor shift based image stabilization implementations are capable of correcting camera roll rotation. A disadvantage compared to EIS is that shift modules tend to be larger and as a result more expensive to implement due to the addition of actuators and the need for power driving sources [Golik, 2006].

2.4 Electronic image stabilization

Electronic image stabilization (EIS) is a stabilization method which uses algorithms for comparing contrast and pixel location between frames. The comparing is done between every frame and the differences are then used to create new frames which suffer less from vibrational motion. The fact that this stabilization method is done by software makes it inexpensive, however, the image quality will always be reduced. Moreover, under low light conditions or at full electronic zoom the EIS will suffer compared to other stabilization methods [ROHM-Semiconductor, 2017].

Some EIS algorithms can reduce computing time if they are provided information about vibrations from an accelerometer and angular velocity from a gyroscope. [Alexandre Karpenko, 2011]

2.5 Mechanic-based image stabilization

In mechanic-based image stabilization the solutions is built around the camera module. One method that is purely mechanical is "steadicam" [Golik, 2006]. Another method is to build housing around the camera and compensate the vibration with the help of actuators. To measure vibration, gyroscopes and accelerometers are used in the same way as some of the prior technologies mentioned. Moreover, the technique can compensate for bigger amplitudes of vibration than the other technologies, but the drawback is the size of the housing with actuators and the need for power driving sources, so it is not suited in small spaces.

3

Control and system overview

This chapter will give an overall view of the system that has been used. Furthermore a description of the control concept and possible disturbances will be presented in the end.

3.1 Platform

The platform that has been used is an Axis PTZ-camera (pan-tilt-zoom), model number Q6155-E. The platform is mounted so that the camera dome is under the camera housing, as shown Figure 3.1. Pan can move from 0° to 360° and then start over from 0° thanks to the slip ring construction. The tilt is limited to 20° (above horizon) and -90° (below horizon). This limit comes from the dome construction.

$$\phi_{Pan} \quad (Limitless) \quad (3.1)$$

$$-90^\circ \leq \theta_{Tilt} \leq 20^\circ \quad (At \ horizon) \quad (3.2)$$



Figure 3.1 Axis PTZ-camera model Q6155-E [Axis, 2017]

3.2 Actuators, motor drivers & motion controllers

The Q6155-E camera model has stepper motors as actuators for pan and tilt respectively. The stepper motors do not directly drive the camera module but there is a belt gear between them. Stepper motor drivers constitute the power electronics that drives the stepper motors, and they are configured in micro-step mode to get a better precision. Moreover a motion controller which is a trajectory generator for velocity and position is used for the PTZ-camera. Its task is to make the movement smooth and not so jerky. The pan and tilt speed limits are $700^\circ/s$ and $500^\circ/s$ respectively.

$$-700^\circ/s \leq \frac{d\phi_{Pan}}{dt} \leq 700^\circ/s \quad (3.3)$$

$$-500^\circ/s \leq \frac{d\theta_{Tilt}}{dt} \leq 500^\circ/s \quad (3.4)$$

3.3 Sensors

Three different types of sensors in the PTZ-camera have been used in this thesis. These are:

- Inertial measurement unit (IMU), consisting of a three-axis gyroscope and an accelerometer
- Pan and tilt encoders

The IMU can be used to measure camera orientation and the encoder provides position feedback regarding pan and tilt.

Inertial measurement unit (IMU)

The IMU integrated in the camera is a model called MPU-6050. It consists of a 3-axis accelerometer and a 3-axis gyroscope. Moreover six 16-bit analog-to-digital converters digitizes the sensor outputs.

Different ranges can be set for the gyroscope and the accelerometer and in this thesis Axis default settings were used, i.e. $\pm 250^\circ/s$ for the gyroscope and $\pm 8g$ for the accelerometer.

Encoders

Two encoders are mounted on the camera module side of the belt gear in the camera. The encoders provide information about the angular position for both pan and tilt by dividing a 360° rotation into discrete steps. A certain sensor value thereby corresponds to a certain angle relative to the origin position.

3.4 System architecture

The system architecture seen in Figure 3.2 describes the motor control and the handling of sensors in the Q6155-E model. The whole system is powered by Power over Ethernet (PoE) and the heart of the system is the Main Central Processing Unit (MCPU) that runs embedded Linux operative system. The MCPU handles all the communication via Ethernet, for example sending video streams and settings. The MCPU also handles data collection from the gyroscope and the accelerometer. This data is today used by the EIS to remove low amplitude vibration with high frequency in the video images. From an HMI (Human Machine Interface) motor commands can be sent via Ethernet to the MCPU, the MCPU then forwards the commands to the MCU (Microprocessor Unit) that executes the task as seen in Figure 3.2. If needed, information about pan and tilt position can be obtained from an encoder. However, the encoder data must be forwarded by the MCU in order to use it in the MCPU. The communication protocol used between the MCU and the MCPU is I²C.

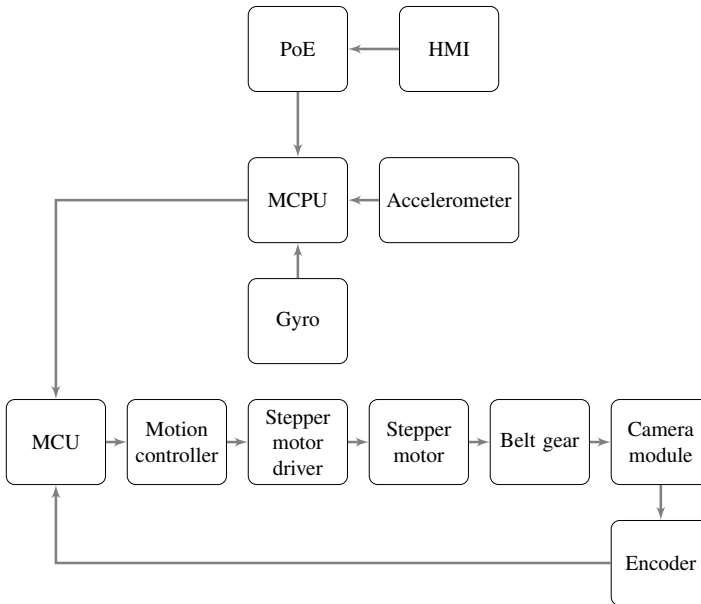


Figure 3.2 System architecture of Q6155-E

3.5 Control concept

The basic idea in the control was that pan and tilt could be controlled independent of each other. This led to the control architecture shown in Figure 3.3. As seen, the two control loops get one reference signal each and both receive encoder feedback. The reference signals come from a reference generator which creates compensation references with the help of data from the encoders and the IMU. A deviation from desired camera orientation will change the reference signals in order to suppress the disturbance.

The controller types that were considered were two PID-controllers. Moreover a model of the camera dynamics was needed for simulation and to obtain good controller parameters.

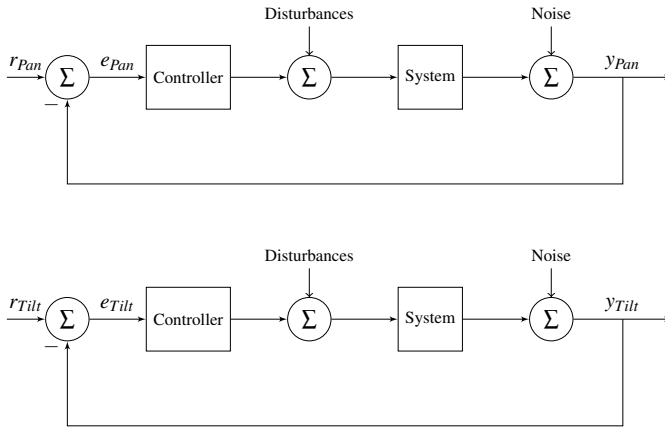


Figure 3.3 Control concept

3.6 Disturbances at sea

Axis has made experiments at sea where IMU-data has been gathered in order to study which disturbance frequencies that usually appear. The boat that was used during the experiments and the camera mounting can be seen in Figure 3.4. Furthermore, some of the collected data can be seen in Figure 3.5. The figure shows integrated gyroscope data and thereby it gives information about how orientation can change at sea. Note, however, that the axis of rotation is not known. As mentioned earlier the goal was to handle disturbance frequencies between 0 and 1Hz and this seems reasonable since the figure shows frequencies with relatively high amplitude in this interval. There are, however, frequency components with higher frequencies but with smaller amplitude as well, but these are easier to handle with the EIS.



Figure 3.4 Camera mounting during experiments at sea

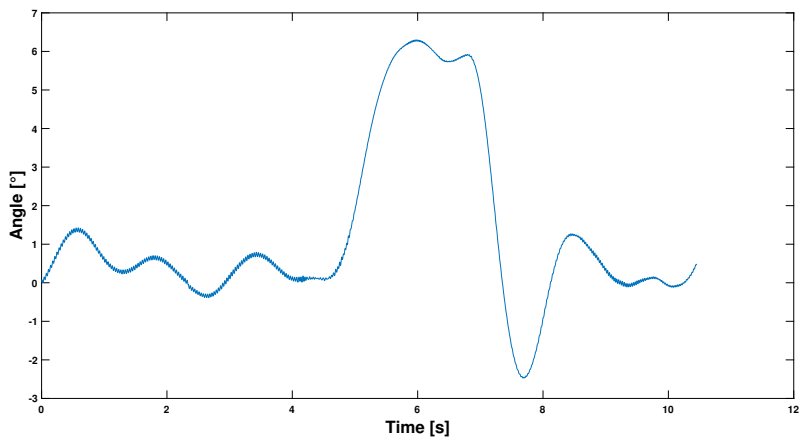


Figure 3.5 Example of integrated gyroscope data at sea

4

Orientation and sensor fusion

The first problem that had to be solved in the thesis was how to obtain the camera orientation from the IMU. This problem lead into the area of sensor fusion and two different methods for estimating orientation, the Kalman and Complementary filter. The performance of them both was tested and compared in order to implement the best one.

This chapter will describe the whole process of orientation estimation, starting out by introducing necessary coordinate systems and ending with the presentation of real orientation data.

4.1 Coordinate systems

In order to keep the camera stabilized in a fixed position it is important that the camera position can be measured relative to something that is fixed, namely a fixed reference frame (inertial frame). In this thesis the North-East-Down (NED) reference frame has been used for orientation, where "down" is aligned with the direction of gravity [CHRobotics, 2012].

The sensor measurements, however, cannot always be measured directly in the inertial frame because the sensors measure quantities relative to their own frame of reference, which is not fixed in space, see Figure 4.1. This means that a conversion of measurement data has to be done to get an inertial frame representation. A way to do this is to use Euler angles as a representation of how the sensor frame is oriented relative to the inertial frame [CHRobotics, 2012].

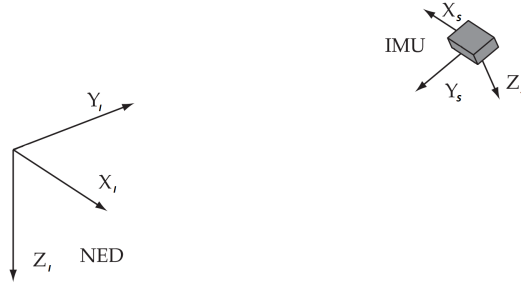


Figure 4.1 Inertial reference frame relative to unfixed sensor frame [Alves Neto et al., 2009]

Euler angles

As mentioned above Euler angles can be used for describing the orientation of an inertial frame relative to a moving frame. The angles are denoted ψ , θ and ϕ and represents the result of three rotations about different axes [CHRobotics, 2012]. The rotations are called yaw, pitch and roll and can be represented as rotation matrices as seen in Equations 4.1–4.3. The lower and upper indicis represent the starting and ending frame for the rotation. To get a rotation matrix representing all three rotations, all three rotation matrices have to multiplied, but one has to be careful, because the order of the multiplication matters. In this thesis the so-called aerospace rotation sequence will be used [Pedeley, 2013]. In this sequence the order is yaw, pitch, roll. This means that the the yaw rotation matrix rotates a vector from the inertial frame (I) to what in this thesis is called the first intermediate frame ($In1$). The pitch rotation matrix rotates from $In1$ to the second intermediate frame ($In2$) and lastly the the roll rotation matrix rotates from $In2$ to the sensor frame (S). This is expressed mathematically in Equation 4.4 and the final rotation matrix for the aerospace rotation sequence can be seen in Equation 4.5, where c and s represents cosine and sine respectively.

$$\mathbf{R}_I^{In1}(\psi) = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

$$\mathbf{R}_{In1}^{In2}(\theta) = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (4.2)$$

$$\mathbf{R}_{In2}^S(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \quad (4.3)$$

$$\mathbf{R}_I^S(\phi, \theta, \psi) = \mathbf{R}_{In2}^S(\phi) \mathbf{R}_{In1}^{In2}(\theta) \mathbf{R}_I^{In1}(\psi) \quad (4.4)$$

$$\mathbf{R}_I^S(\phi, \theta, \psi) =$$

$$\begin{pmatrix} c(\theta)c(\psi) & c(\theta)s(\psi) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & c(\phi)c(\psi) + s(\theta)s(\phi)s(\psi) & c(\theta)s(\phi) \\ c(\phi)c(\psi)s(\theta) + s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{pmatrix} \quad (4.5)$$

Equation 4.5 can be used for estimating the pitch and roll angles from accelerometer readings which is described in Section 4.2 [Pedeley, 2013]. Gyro readings however, cannot be represented in the inertial frame using 4.5. For this the rotation matrix in Equation 4.6 is needed. The axes of the IMU-sensor frame in this thesis are denoted X_s , Y_s and Z_s and when the camera is horizontal X_s is in the opposite direction of gravity. Z_s is in the optical axis direction and Y_s is perpendicular to them both. So if the angular velocities in the sensor frame are denoted x_{vel} , y_{vel} and z_{vel} the Euler angle rates can be calculated as in Equation 4.7 [CHRobotics, 2012].

$$\mathbf{D}(\phi, \theta, \psi) = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{pmatrix} \quad (4.6)$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{pmatrix} \begin{pmatrix} z_{vel} \\ y_{vel} \\ x_{vel} \end{pmatrix} \quad (4.7)$$

An important thing to notice about Equation 4.7 is that a pitch angle θ of 90° will cause some matrix elements to diverge towards infinity. This is a phenomenon called gimbal lock which sets a limitation for Euler angles. This leads to the conclusion that Euler angles should not be used in applications where the pitch angle will get near $\pm 90^\circ$ [CHRobotics, 2012].

4.2 Attitude estimation with accelerometers

Accelerometers can be used for both sensing linear acceleration and strength of the gravitational field. The latter can be used for estimating attitude relative to the ground, but only under static conditions, i.e. no linear acceleration. To do this Equation 4.5 can be used as it describes the relationship between the inertial and sensor frame vector representation [Pedeley, 2013]. Under static conditions the relation looks as follows:

$$\begin{pmatrix} Z_s \\ Y_s \\ X_s \end{pmatrix} = \mathbf{R}_I^S(\phi, \theta, \psi) \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (4.8)$$

where -1 represents -1g. The minus sign is necessary in order to get alignment between the sensor frame and inertial frame when all Euler angles are zero, i.e. no rotation has been done. However, this only holds when the camera is mounted with the chassis upwards, see Figure 3.1. In the opposite case the -1g needs to be replaced with just 1g. This is however not how the camera is supposed to be mounted. Moreover, the vector elements in the sensor frame are also represented in the unit g. If the multiplication in Equation 4.8 is carried out the following is obtained:

$$\begin{pmatrix} Z_s \\ Y_s \\ X_s \end{pmatrix} = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta)\sin(\phi) \\ -\cos(\theta)\cos(\phi) \end{pmatrix} \quad (4.9)$$

Solving for θ and ϕ gives:

$$\phi = \text{atan2}\left(-Y_s, -X_s\right) \quad (4.10)$$

$$\theta = \text{atan2}\left(Z_s, \sqrt{Y_s^2 + X_s^2}\right) \quad (4.11)$$

where $\text{atan2}(y, x)$ returns the arcus tangent of y/x with respect to sign of the input parameters. Thereby the right quadrant is determined [Bilting and Skansholm, 2011].

The calculations above show that only the pitch and roll angles can be estimated using an accelerometer and the aerospace rotation sequence. This comes from the

fact that a yaw rotation does not change the static measurements of the accelerometer in any way. It is also important to remember that the calculations do not hold when linear acceleration is present. However, in this thesis, estimation has to be done also when the camera is undergoing linear acceleration. This means that the measurements from the accelerometer has to be complemented in some way. Fortunately it was found that the gyroscope can do just that.

4.3 Sensor fusion

The idea of sensor fusion is to combine measurements from several sensors in order to get more reliable information. Because of the unreliable attitude estimation described in Section 4.2 it seemed reasonable to investigate if it was possible to fuse measurements from the gyroscope and the accelerometer and thereby improving the estimation accuracy. Two different fusion algorithms were looked into, the Complementary filter and the Kalman filter.

Complementary filtering

When measuring attitude with gyroscopes and accelerometers each sensor has its disadvantages. The accelerometer, as described above, does not provide reliable measurements when the attitude is changing and the gyroscope suffers from a time-varying bias. A way around this problems is to combine the best part of each sensor to obtain an estimate that better corresponds to the real world. A fairly simple but yet effective way to do this is to use the complementary filter [Zhi, 2016]. In the frequency domain it can be described by

$$\theta(s) = \frac{1}{1 + \tau s} A(s) + \frac{\tau s}{1 + \tau s} \frac{1}{s} \Omega(s) = \frac{A(s) + \tau \Omega(s)}{1 + \tau s} \quad (4.12)$$

where $A(s)$ is the frequency domain representation of accelerometer signal, $\Omega(s)$ the gyroscope signal, $\theta(s)$ the attitude angle and τ the time constant of each filter. So the idea with the complementary filter is to low-pass filter the data provided by the accelerometer, integrate and high-pass filter the data from the gyroscope and then add them up to give a more reliable attitude estimate. In this way the effect that attitude changing has on the accelerometer is filtered out and complemented with gyroscope data. The time-varying bias of the gyroscope is also filtered out and complemented by accelerometer data. So in short, the filter provides an estimate that is based on the best measurements of each sensor. Figure 4.2 shows the Bode plot of both the high- and low-pass filter. It can easily be seen that the complementary filter can operate over larger frequency band than each sensor for itself [Zhi, 2016] [Higgins, 1975].

A problem with Equation 4.12 is that it is in continuous time and can thereby

not be implemented on a computer. To discretize it, backward difference can be used. Equation 4.12 then becomes

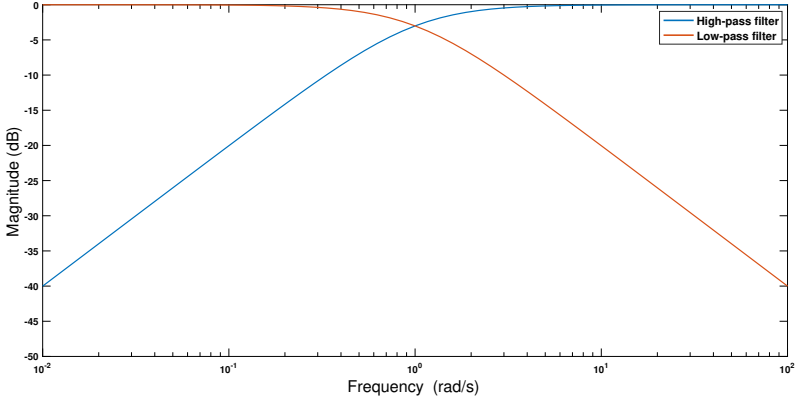


Figure 4.2 Bode plot of high and low-pass filter that together constructs the complementary filter, here with $\tau = 1$

$$s = \frac{z-1}{zh} \quad \Rightarrow \quad \theta(z) = \frac{(A(z) + \tau\Omega(z))h}{(h+\tau) - \tau z^{-1}} \quad (4.13)$$

where h is the sampling interval. Conversion to the time domain gives:

$$\theta(k) = \frac{\tau}{h+\tau}(\theta(k-1) + \omega(k) \cdot h) + \frac{h}{h+\tau} \cdot a(k) \quad (4.14)$$

Setting $\alpha = \tau/(h+\tau)$ gives:

$$\theta(k) = \alpha(\theta(k-1) + \omega(k) \cdot h) + (1-\alpha) \cdot a(k) \quad (4.15)$$

Equation 4.15 can easily be implemented on a computer, and moreover its simplicity sets little requirements on computational capacity. In that sense it is a good alternative to more complex filters, such as the Kalman filter [Zhi, 2016].

In this thesis the filter has been implemented according to Equation 4.15 but note that the filter does not work directly with raw sensor data. Some processing has been done before the filtering, see Figure 4.3. In order to obtain an Euler angle from the accelerometer trigonometry has to be used, see Section 4.2. The gyroscope

data also has to be processed to get an earth frame representation, see Equation 4.7. Note, in the figure, that this is done by using Euler angles estimated in the previous sampling interval. Moreover, it needs to be integrated to get an Euler angle that can be used by the filter.

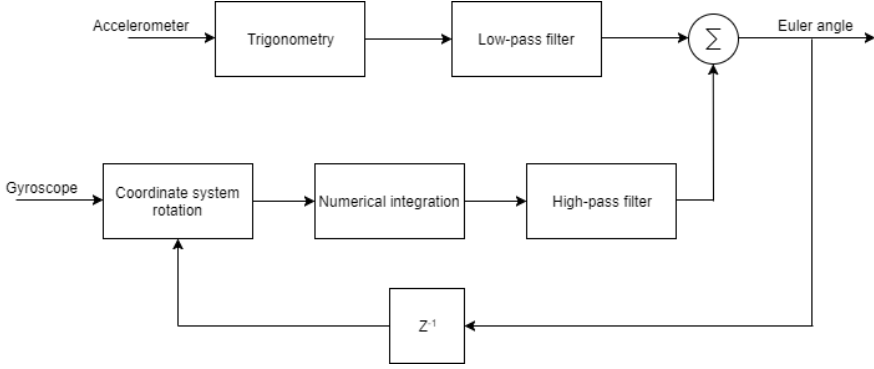


Figure 4.3 Block diagram of the complementary filter, including raw data processing

Kalman filtering

The Kalman filter is an iterative algorithm which combines noisy measurements and predictions to provide an optimal estimate of the state of a dynamical system. That the filter is optimal refers to the fact that it minimizes the estimation error covariance. However, for the filter to work a mathematical model for prediction is needed and moreover the noise covariance has to be estimated and included in the algorithm. In short the algorithm can be described as a predictor-corrector algorithm. This means that predictions are made by using the dynamical model and these predictions are then "corrected" by measurements. So the measurements acts as a feedback for the prediction [Elmenreich, 2002]. The dynamical model that is needed can if linear be described by

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \quad (4.16)$$

where \mathbf{x} is the state vector, \mathbf{A} the state transition matrix, \mathbf{B} the input matrix, \mathbf{u} the input vector and \mathbf{w} the process noise. The process noise is modeled as white noise meaning $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ where \mathbf{Q}_k is the covariance matrix [Elmenreich, 2002]. Another equation is also needed in order to relate the measurements to the states, which is given by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \quad (4.17)$$

where \mathbf{C} is the observation matrix, \mathbf{y} the sensor measurements and \mathbf{v}_k the measurement noise. The measurement noise can also be described as white noise $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$ with covariance matrix \mathbf{R}_k . With the equations above the prediction part of the algorithm can be described as follows:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \quad (4.18)$$

$$\mathbf{P}_{k+1} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q}_k \quad (4.19)$$

where Equation 4.18 predicts the future state vector \mathbf{x}_{k+1} and the second, Equation 4.19, predicts the estimation error covariance of the state. The next step of the algorithm is the update, where the predictions are corrected by measurements. This is done in the following way:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k+1}\mathbf{C}^T + \mathbf{R}_{k+1})^{-1} \quad (4.20)$$

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}\hat{\mathbf{x}}_{k+1}) \quad (4.21)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C})\mathbf{P}_{k+1} \quad (4.22)$$

where \mathbf{K} is known as the Kalman gain. This gain is what minimizes the state estimation error when converged to a stationary value [Glad and Ljung, 2003]. Equations 4.18–4.22 contain all steps of the algorithm and after 4.22 it is repeated. This iterative behavior and the fact that every iteration takes approximately the same time makes the algorithm well-suited for real-time applications [Elmenreich, 2002].

In order to predict the pitch and roll angle, using the gyroscope and Equation 4.7 only, the following dynamical model was considered:

$$\theta_k = \theta_{k-1} + (\dot{\theta}_k - b_{k-1})h \quad (4.23)$$

where θ represents the pitch angle, $\dot{\theta}$ the pitch rate, b the bias of the pitch rate and h the sampling interval of the IMU [Sloth Lauszus, 2012]. The same model can be applied to predict the roll and yaw angle.

In order to get a state space representation like the one presented in Equation 4.16 the following state vector can be chosen:

$$\mathbf{x} = \begin{pmatrix} \theta \\ b \end{pmatrix} \quad (4.24)$$

leading to the state space representation

$$\begin{pmatrix} \theta_k \\ b_k \end{pmatrix} = \begin{pmatrix} 1 & -h \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{k-1} \\ b_{k-1} \end{pmatrix} + \begin{pmatrix} h \\ 0 \end{pmatrix} \dot{\theta}_k + \mathbf{w}_k \quad (4.25)$$

where the noise is assumed to be uncorrelated meaning that the covariance matrix \mathbf{Q}_k looks as follows

$$\mathbf{Q}_k = \begin{pmatrix} Q_\theta & 0 \\ 0 & Q_b \end{pmatrix} \quad (4.26)$$

where Q_θ and Q_b represents the variance of the noise that is corrupting the estimation of θ and b respectively. In this case the noise in θ comes from the gyroscope measurements [Pycke, 2006]. Due to this fact that variance could be found by experiment.

In the update step of the algorithm the accelerometer can be used to correct the measurements made by the gyroscope [Sloth Lauszus, 2012]. This is done by using the accelerometer as described in Section 4.2. The relation between the measurements and the the states according to Equation 4.17 is then given by

$$y_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \theta_k \\ b \end{pmatrix} + v_k \quad (4.27)$$

where v is a scalar meaning that the measurement noise covariance matrix \mathbf{R}_k is just the variance of the measurement noise, i.e. the accelerometer noise. This value could also be found experimentally.

Note that only the roll and pitch angles could be estimated with the Kalman filter. Although the yaw angle could be predicted it could never be corrected by the accelerometer. The same holds for the Complementary filter. This was a problem that had to be solved in order to get the reference generator working.

To clarify it can also be mentioned that one Complementary/Kalman filter either estimates roll or pitch. To obtain both, two filters are needed.

Figure 4.4 shows a block diagram of how the Kalman filter was implemented in this thesis. As can be seen the implementation was done in the same context as the Complementary filter.

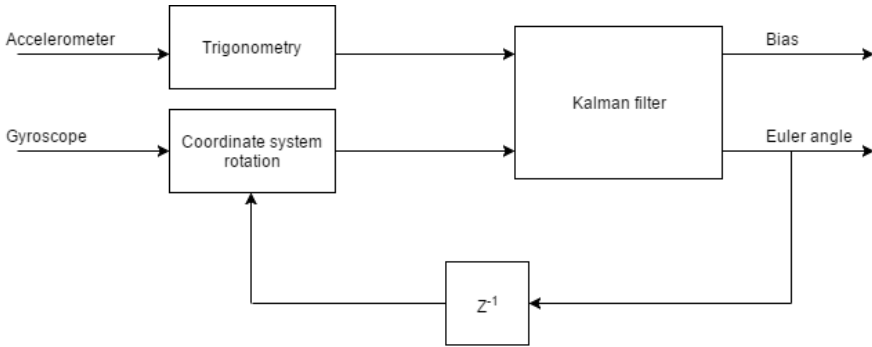


Figure 4.4 Block diagram of the Kalman filter, including raw data processing

4.4 Result and discussion

Separate use of gyroscope and accelerometer

To see the disadvantages of the attitude estimation using the accelerometer a test using only the accelerometer was carried out. The result can be seen in Figure 4.5. Note that the estimation was done under static conditions although noise with a standard deviation of $0.0098\text{rad} \approx 0.56^\circ$ is present. The experiment shows that the accelerometer is corrupted with short-term fluctuations even under static conditions. To remove this a low-pass filter can be used, and this is exactly what the Complementary filter provides.

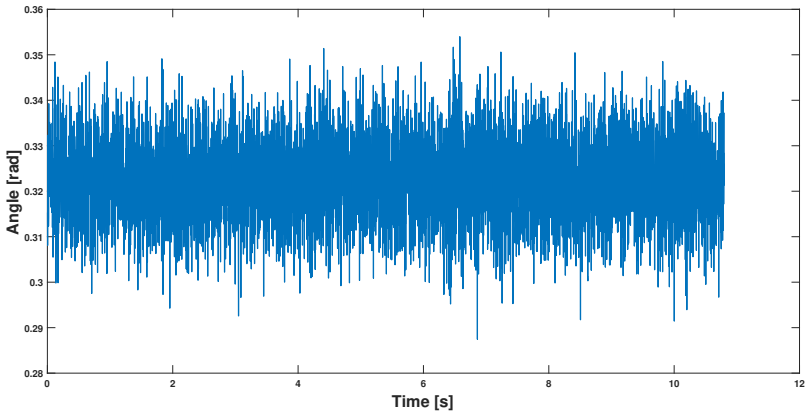


Figure 4.5 Accelerometer estimation of pitch

Figure 4.6 also shows an estimation of the pitch angle, the difference is that the estimation is done by integrating the rotated gyroscope data. Note that the scale on the y-axis is the same as in Figure 4.5 but the stationary value is different due to gyroscope bias and an incorrect initial pitch value. Remember that the pitch value can only change during motion if a gyroscope is used. However, the interesting thing to look at is the stationary behavior which differs significantly from the accelerometer. It is clear that the gyroscope does not suffer from short-term fluctuations, but rather from long time drift, which is a result of an accumulation of errors in the pitch rate measurement. To remove this a high-pass filter can be used which also is provided by the Complementary filter.

Note that similar results were obtained when estimating the roll angle.

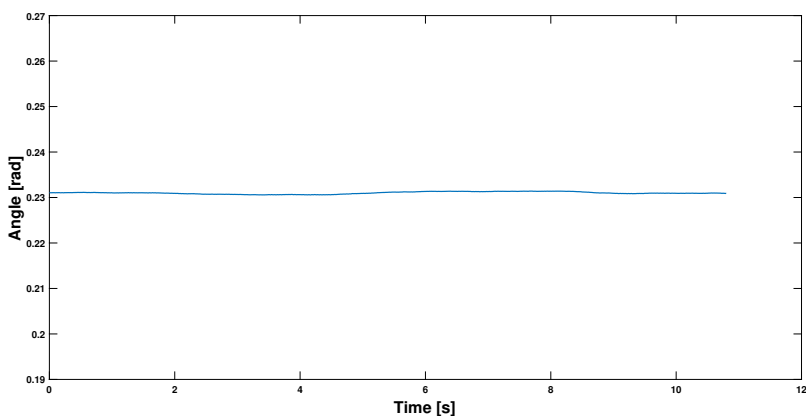


Figure 4.6 Gyroscope estimation of pitch, i.e. integration of pitch rate

Complementary and Kalman filtering

In order to decide which filter to use for the sensor fusion, experiments were carried out with both filters. The experiments were carried out under both static and dynamic conditions.

In Figure 4.7 the result of both filters are shown. This is for the static case and an initial pitch angle equal to zero. It can be seen that the Complementary filter converges faster than the Kalman filter which is partly a consequence of the choice of initial estimation error covariance. In stationarity, however, the Kalman filter suffers less from short-term fluctuations which most certainly is a consequence of a better statistical description of the noise compared to the Complementary filter where the filter is obtained by choosing a cut-off frequency only [Higgins, 1975].

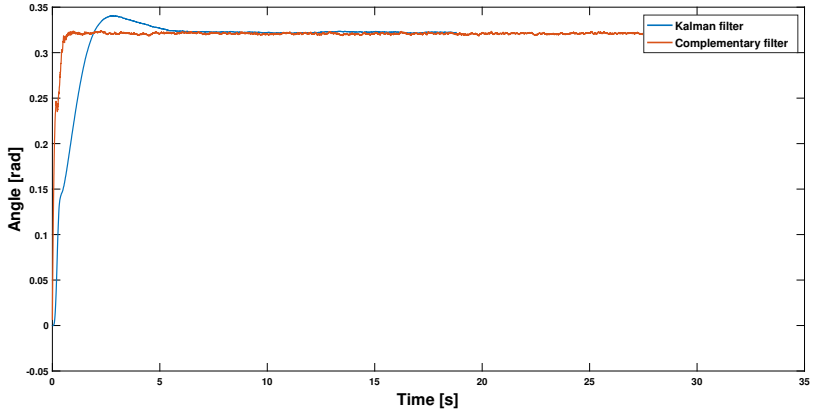


Figure 4.7 Kalman and Complementary filter estimation of pitch, under static conditions

Figure 4.8 shows the result of the experiment carried out under dynamic conditions, i.e. a changing pitch angle. It can be seen that the Kalman filter follows the changing pitch angle in the same way as the Complementary filter. Moreover, from Figure 4.9, it is clear that the Complementary filter suffers more from short-term fluctuations even in this case.

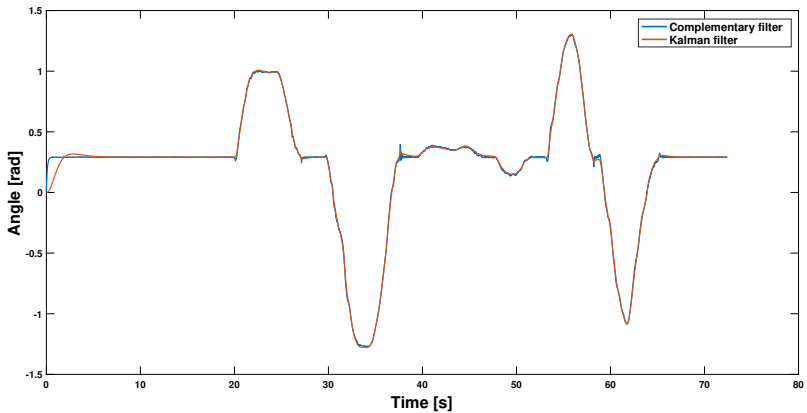


Figure 4.8 Kalman and Complementary filter estimation of pitch under dynamic conditions

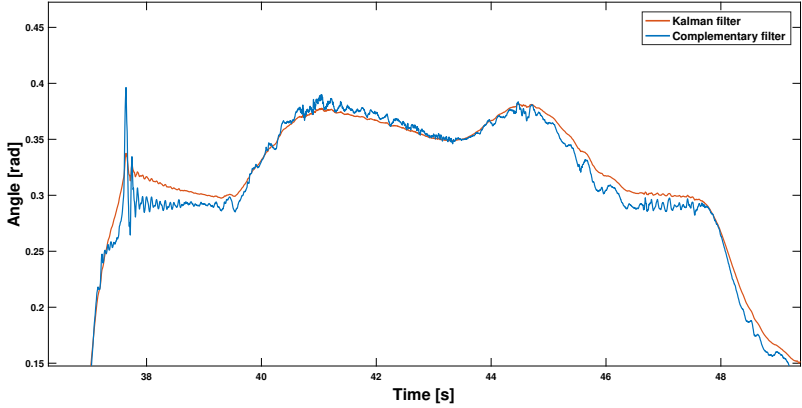


Figure 4.9 Kalman and Complementary filter estimation of pitch under dynamic conditions (Figure 4.8 enlarged)

4.5 Conclusion

From the results it is clear that the sensor fusion gives more reliable information about the current orientation. Moreover, the Kalman filter shows a less noisy estimation than the Complementary filter, which gives reason for choosing the Kalman filter in a final implementation. However, the Kalman filter requires more computation time but this is negligible since the sampling interval was chosen long enough.

Another advantage of the Kalman filter is that it gives information about the bias of the roll and pitch rates. This can be seen by plugging in the update of the bias in Equation 4.21. The update is then given by:

$$b_k = \hat{b}_k + K_k(\theta_k - \hat{\theta}_k) \quad (4.28)$$

which means that the bias is obtained by integrating the error between the accelerometer reading and the prediction made by the gyroscope. Due to the fact that the bias can change over time, mainly because of temperature changes, the Kalman filter has a clear advantage over the Complementary filter and because of this the Kalman filter was implemented.

There was one major problem with the orientation estimation, the yaw angle. As described earlier it cannot be obtained by sensor fusion due to the accelerometer. This led to the solution of just integrating the yaw rate given by the rotated gyroscope data. The integration is given by:

$$\psi = \int_{t_0}^t \frac{d\psi}{d\tau} d\tau \quad (4.29)$$

This means that the yaw angle always will be given the value zero initially, regardless of true orientation relative to the north-east direction. Another issue is the bias which will not be estimated continuously. To solve this a yaw rate bias estimation was implemented to be carried out under static conditions. The idea of the estimation was to gather as much yaw rate data as possible under a certain time and then take the mean value of it in order to obtain a bias estimate. This gave a decent result, meaning that the drift was reasonable. However, under changing temperatures it might be worse.

The only way to solve the problem regarding true orientation is to provide additional data to the camera, from a magnetometer for example. This data can then be fused with the gyroscope in order to get a more reliable estimate. However, the integration was enough to prove the concept of the thesis.

5

Reference generation

The estimation of the camera orientation opened up for the opportunity to create a feedback path for the upcoming control design. This was the initial thought, but this was later reconsidered since the camera actuators do not control pitch and yaw directly, unless the camera chassi is horizontal. The actuators do rather control pan and tilt which is the camera orientation relative its own chassi and not the NED-frame. This fact lead to the idea of reference generation, which means that reference signals for pan and tilt would be created in order to reach the desired NED-frame orientation. To do this two new frames had to be introduced resulting in a total of four frames, described as follows:

- NED-frame - North-East-Down inertial frame
- Sensor frame - frame which follows the movements of the sensor.
- Desired frame - frame which is oriented in the desired camera orientation relative to the NED-frame.
- Chassi frame - frame which follows the movements of the camera chassi and is thereby an inertial frame for pan and tilt motions.

With the frames defined the problem could be stated as follows: How does the sensor frame need to be oriented relative to the Chassi frame in order to align with the desired frame?

The first step of the solution requires a rotation matrix which describes the rotation that has to be done relative to the Chassi frame. This can be expressed as follows:

$$\mathbf{R}_{Chassi}^{Desired} = \mathbf{R}_{NED}^{Desired} \mathbf{R}_{Sensor}^{NED} \mathbf{R}_{Chassi}^{Sensor} \quad (5.1)$$

where the matrices on the left have the same look as in Equation 4.5 and its inverse, which means that three sets of Euler angles have to be known in order to

get a matrix containing only numerical values. The angles that create the matrices are the following:

- $\mathbf{R}_{NED}^{Desired}$ - matrix defined by the desired Euler angles relative to the NED-frame $(\phi_d, \theta_d, \psi_d)$, which is given by the user.
- $\mathbf{R}_{Sensor}^{NED}$ - matrix defined by the current Euler angles of the sensor frame relative to the NED-frame (ϕ, θ, ψ) , i.e. the angles estimated with IMU-data.
- $\mathbf{R}_{Chassi}^{Sensor}$ - matrix defined by the current Euler angles of the sensor frame relative to the Chassi frame $(\phi_c, \theta_c, \psi_c)$, where ψ_c and θ_c is given by the pan and tilt encoders. ϕ_c is always equal to zero since a roll rotation cannot be done relative to the Chassi frame.

This means that $\mathbf{R}_{Chassi}^{Desired}$ can be created at any time. Furthermore, if it is possible to find Euler angles relative to the Chassi frame that directly creates $\mathbf{R}_{Chassi}^{Desired}$ these can be used as references for pan and tilt. Obviously, Euler angle extraction is then needed.

5.1 Extracting Euler angles from a rotation matrix

The extraction of Euler angles can be done by first expressing $\mathbf{R}_{Chassi}^{Desired}$ as an arbitrary rotation matrix given by

$$\mathbf{R}_{Chassi}^{Desired} = \begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} \quad (5.2)$$

and then match the matrix elements in Equation 4.5 with it [Day, 2012]. The roll angle can be extracted in the following way:

$$\phi_c = \text{atan2}(R_{12}, R_{22}) = \text{atan2}(s(\phi_c)c(\theta_c), c(\phi_c)c(\theta_c)) \quad (5.3)$$

Moreover the pitch angle can be extracted by first computing:

$$\cos(\theta_c) = \sqrt{R_{00}^2 + R_{01}^2} \quad (5.4)$$

and then

$$\theta_c = \text{atan2}(-R_{02}, c(\theta_c)) \quad (5.5)$$

Lastly ψ_c can be obtained by inserting the pitch and roll angles into Equation 4.5 with $\psi_c = 0$. The transpose of the resulting matrix can then be multiplied with $\mathbf{R}_{Chassi}^{Desired}$ to get a result which can be matched with Equation 4.1. The matching results in:

$$\psi_c = \text{atan2}(s(\phi_c)R_{20} - c(\phi_c)R_{10}, c(\phi_c)R_{11} - s(\phi_c)R_{21}) \quad (5.6)$$

which completes the Euler angle extraction [Day, 2012].

The calculations made make it possible to use ψ_c as a reference for pan and θ_c as a reference for tilt since those values will result in an alignment of the sensor frame and the desired frame. Note, however, that the angle ϕ_c cannot be used as a reference in this thesis due to the fact that the camera cannot do a roll rotation relative to the chassi frame. This means that the desired orientation relative to the NED-frame is not possible to reach in all cases.

5.2 Result and discussion

In order to test the reference generation a Simulink simulation was carried out under the following conditions:

- $(\phi_d, \theta_d, \psi_d) = (0, 0, 0)$
- $(\phi, \theta, \psi) = (0, P_d(t), 0)$, i.e. a time varying pitch disturbance

Meaning that only a tilt reference was generated. Three different time varying pitch disturbances were investigated, namely:

- $P_d(t) = 0.3 \cdot \sin(\pi t)$, with amplitude in radians
- $P_d(t) = 0.8 \cdot H(t - 1)$, where $H(t - 1)$ is the Heaviside step function
- $P_d(t)$ has the shape of a triangle wave with amplitude 1rad and frequency 0.5Hz

The results of the three cases can be seen in Figures 5.1 - 5.3. The figures show clearly that the reference generator produces a tilt reference which suppresses the disturbance completely. In reality however, there will be time delays. The computational time delay for the reference generation was measured to be around 0.7ms. Moreover, the sampling interval was set to 50ms meaning that the computational delay could be considered negligible. The reason for this choice was that the final system could not operate at a faster rate. Another time delay of importance that had to be investigated was the process time delay. This is described in more detail in Sections 6 and 8.

As described in Equation 3.2 in Section 3 the tilt motion of the camera has a physical limit. This limit was implemented in the reference generator to not exceed the mechanical limitation of the camera tilt.

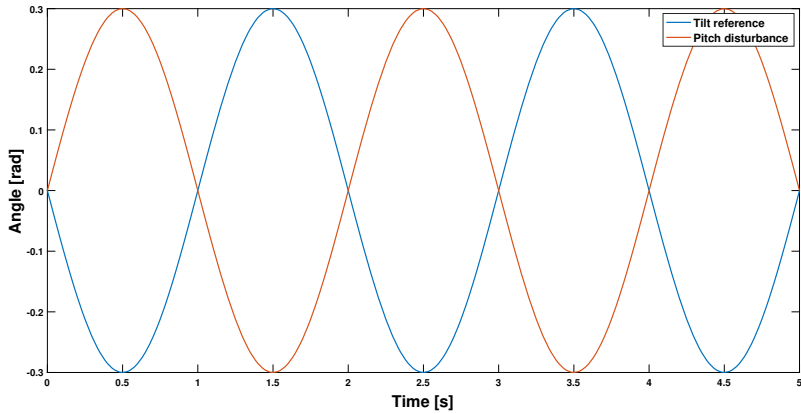


Figure 5.1 Tilt reference generation when pitch disturbance is sinusoidal

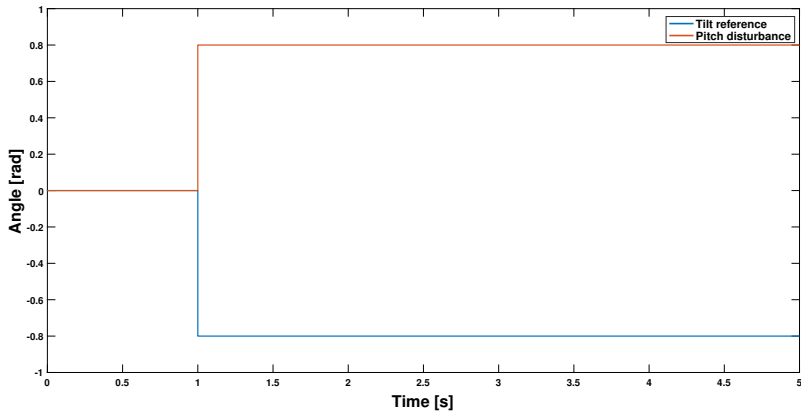


Figure 5.2 Tilt reference generation when exposed by a pitch step disturbance

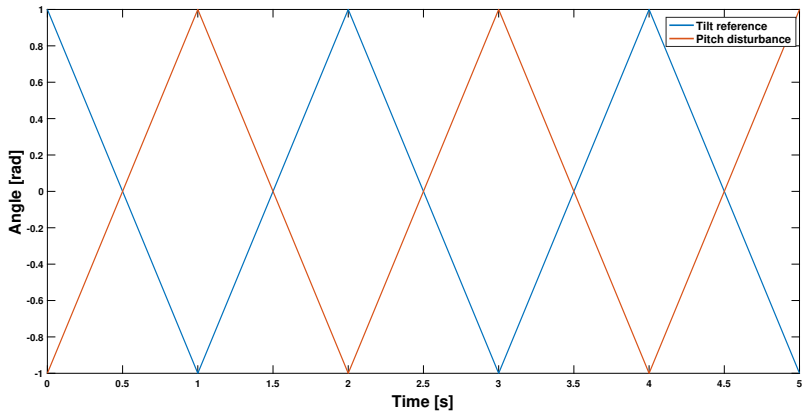


Figure 5.3 Tilt reference generation when exposed by a triangle wave pitch disturbance

6

Modelling

A model of the camera was needed for simulation and for finding appropriate control parameters for the controller. In this thesis this was done in three ways. The first approach was to model the camera with a psychical model, i.e., with the laws of nature. However, this was never used in the control design due to the extensive work of finding the right physical parameters and also due to the fact that it was actually the motion controllers that decided the behavior of camera movements.

This realization lead to the idea of modelling the camera with motion profiles, which is what is used by the motion controller. This approach was however not used in the final control design, due to lack of time. The idea will be presented in Chapter 7 though.

The third approach was to model the system with system identification. This was done successfully and was later used in the control design.

In the sections below the physical modelling and the system identification is described. The motion profile is as mentioned described in Chapter 7.

Even though the physical model and the motion profile was not used to reach the final result of the thesis they are most relevant to describe since this can be used in later projects which might go deeper into this.

6.1 Physical modelling

The Axis camera uses stepper motors for driving both pan and tilt. However, the stepper motors do not drive the mechanical system directly. In between there are belt gears. Due to this the physical modelling was divided into three parts as seen in Figure 6.1. For the mechanical system block a two-axis gimbal model was investigated. In the three upcoming subsections mathematical models for each subsystem will be presented.

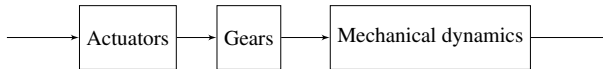


Figure 6.1 Physical model of the system

Stepper motors

The Axis camera model uses stepper motors as actuators for moving pan and tilt. There are three main types of stepper motors:

- Permanent magnet stepper motors
- Hybrid stepper motors
- Variable reluctance stepper motor

The Axis camera uses hybrid stepper motors and as the name applies it is a combination of a permanent magnet and a variable reluctance stepper motor. In Figure 6.2 it can be seen that the windings create the stator poles and that a permanent magnet is mounted on the rotor. Each pole has between two and six teeth and most commonly there are eight stator poles. The windings are two in total and each winding is placed on four of the eight stator poles: winding A is placed on poles 1, 3, 5, 7 and winding B is placed on poles 2, 4, 6, 8. Successive poles of each winding are wound in the opposite sense.

The rotor is a cylindrical permanent magnet, magnetized along the axis with radial soft iron teeth.

The main principle of the motor is that when certain poles generate magnetic flux, the rotor will move until the airgap reluctance of the flux path is minimized [Acar-nely, 2002].

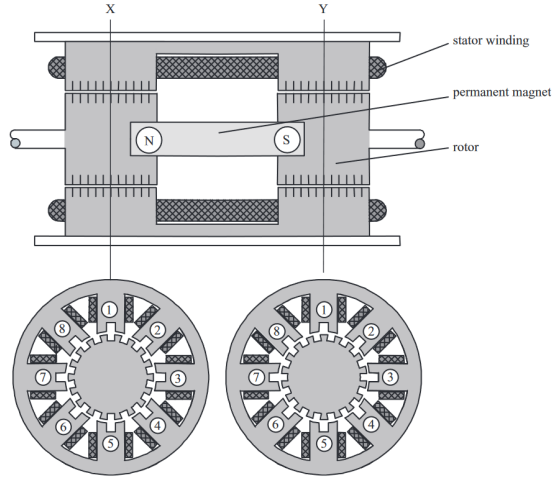


Figure 6.2 Side view and cross-sections of the hybrid stepping motor [Acarnely, 2002]

Mathematical model

For a hybrid stepper motor the dynamics can be described with Equations 6.1–6.4 and the same equations can be found in a number of publications [Kępiński et al., 2015][Acarnely, 2002]. The four equations are non-linear ordinary differential equations:

$$\frac{di_A}{dt} = \frac{1}{L} [V_A - Ri_A + K_m \omega \sin(N_r \Theta)] \quad (6.1)$$

$$\frac{di_B}{dt} = \frac{1}{L} [V_B - Ri_B + K_m \omega \cos(N_r \Theta)] \quad (6.2)$$

$$\frac{d\omega}{dt} = \frac{1}{J} [-K_m i_A \sin(N_r \Theta) + K_m i_B \cos(N_r \Theta) - B\omega - \tau_l] \quad (6.3)$$

$$\frac{d\Theta}{dt} = \omega \quad (6.4)$$

where i_A , i_B , V_A , V_B are currents respectively voltages in phases A and B. R and L are winding resistance and inductance. K_m is the motor torque constant, B is a viscous friction coefficient, N_r is the number of rotor teeth, J is the rotor moment of inertia, τ_l is the load torque, ω is the rotor speed and Θ is its angular position.

Equations 6.1–6.2 determine the current in the motor windings, i.e. phase A and B. Equations 6.3–6.4 correspond to Newton's 2nd law of rotational motion.

Belt Gear

A belt gear is placed between the stepper motors and the two-axis gimbal and does thereby add more dynamics to the system. To transform angular velocity (ω_n) and torque (M_n) from the stepper motor side to the gimbal side the following equations can be used:

$$M_1 = \pm r M_2 \quad (6.5)$$

$$\omega_2 = \mp r \omega_1 \quad (6.6)$$

Where r is ratio of circumference between the belt wheels, M_1 the torque of wheel 1 and ω_1 the angular velocity of wheel 1. The same notation is used for wheel 2 but with the index 2 instead [Ljung, 2007].

Two-axis gimbal system

The equations of motion for the gimbal system were find in the article [Abdo et al., 2013]. Due to the comprehensive calculations carried out in the article only a brief derivation and the resulting equations will be presented here. Furthermore the model was assumed to have no dynamic unbalance, i.e., symmetrical mass distribution.

In Figure 6.3 the gimbal is presented together with three coordinate systems fixed to the platform, body A and body B. The platform frame (i, j, k) is the same as the previously mentioned chassi frame, the B-frame (n, e, k) represents pan motion relative to the platform and the A-frame (r, e, d) represents tilt motion which is done relative to the A-frame. The A-frame coincides with the sensor frame. Also note that, compared to the article, encoders are used to measure η and ε , and the gyroscope/IMU mounted on body A is as mentioned in Chapter 4 used for estimating orientation relative to the NED-frame.

The basic idea for obtaining the equations of motion is to consider the tilt and pan as rigid bodys. In this way Euler's second law can be used, which states:

$$\mathbf{T} = \mathbf{J} \cdot \frac{d}{dt} \boldsymbol{\omega}(t) + \boldsymbol{\omega}(t) \times (\mathbf{J} \cdot \boldsymbol{\omega}(t)) \quad (6.7)$$

where \mathbf{T} is the torque vector applied and contains applied torques to either the e-axis or the k-axis, i.e. tilt and pan. \mathbf{J} is the inertia matrix in either the A or B-frame and $\boldsymbol{\omega}(t)$ is the angular velocity around every axis in either the A or B-frame. So Equation 6.7 can be used for describing the signal relationship between applied torque and angular position and moreover it describes how the pan and tilt system affect each other. For details see the article [Abdo et al., 2013].

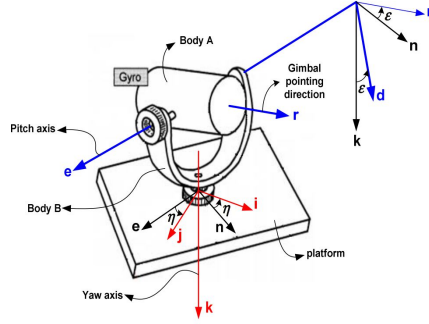


Figure 6.3 Two-axis gimbal system [Abdo et al., 2013]

If the equation is used for the tilt-system without considering any dynamic unbalance the following is obtained:

$$A_e \dot{\omega}_{Ae} = T_e + T_{D1-e} + T_{D2-e} \quad (6.8)$$

where T_{D1-e} and T_{D2-e} are disturbance torques, T_e the applied torque originating from the stepper motor, A_e the e-axis moment of inertia and $\dot{\omega}_{Ae}$ the e-axis angular acceleration.

Since the encoder measures angular position around the e-axis the only thing that has to be done to obtain ε is to divide both sides with A_e and then integrate two times.

The pan-system is a little bit more complicated. Mostly because of the k-axis moment of inertia which changes with movements in the tilt-system. The equation obtained when applying Equation 6.7 is the following:

$$J_{eq} \dot{\omega}_{Bk} = T_k + T_{D1-k} + T_{D2-k} + T_{D3-k} + T_{D4-k} \quad (6.9)$$

where J_{eq} is the k-axis instantaneous moment of inertia, T_k the applied torque, $\dot{\omega}_{Bk}$ the k-axis angular acceleration and the other four terms are disturbance torques. The disturbances do appear due to the cross-coupling between the pan and tilt system.

Likewise the tilt-system η is obtained by dividing both sides with J_{eq} and integrating two times.

The model was investigated in simulation but was never used in the control design as mentioned earlier.

6.2 System identification

System identification deals with the problem of creating dynamical models of systems by investigating the relation between input and output signals. There are several methods for doing this, some more advanced than others. In some cases certain things are already known about the system, for example physical constants. In other cases there are no knowledge about the system. In these cases black box models can be created [Ljung, 2007]. This is the system identification approach that has been applied in this thesis.

Black box model

In contrast to a physical model a black box model only describes the relationship between the input signals and output signals of a system and does not care about the underlying physics [Ljung and Glad, 2004]. There are several methods that can be used to describe the relationship and the method used in this thesis is called step response identification.

Step response identification

Step response analysis is a simple and frequently used method for analyzing how signals affect each other. The idea of the method is to change all input signals at once and then study how the output signals are affected [Ljung and Glad, 2004]. The change of input signals, $u(t)$, is done in the following way:

$$u(t) = u_0, t < t_0; u(t) = u_1, t \geq t_0 \quad (6.10)$$

Meaning that input signals gets the shape of a step. When investigating the effect on the output signals the following can be looked at:

- Time delays
- Static gain
- Character of response, e.g., oscillatory, damped etc.

In this thesis this kind of experiment has been done on two systems, pan and tilt, respectively. Moreover it was assumed that the two systems did not affect each other meaning that two single-input-single-output (SISO) models were created.

To do the experiment a step generator was implemented in the camera software. The signal that was generated was the stepper motor speed signal, giving it the unit *steps/s*, and the output was measured with the gyroscope (*rad/s*). Due to this the experiment was carried out with the sensor frame in alignment with the chassi frame before starting the rotation. This was done to make sure that the gyroscope would measure rotation around the pan- and tilt-axis, i.e., the k- and e-axis in Figure

6.3. The gyroscope was used instead of the encoder since it had a higher sampling rate.

Result and discussion

The results of the pan-experiment is presented in Figure 6.4. As can be seen there are two responses to the step that is done at $t = 0$, the dashed blue line, i.e. the experimental response, and the solid blue line, i.e. the model response. To get these results a model match was done, meaning that a dynamical model with similar step response was chosen to represent the real pan-system. The resulting model can be seen in Figure 6.5. The best match was obtained when the parameters a and T was set to 35 and 0.025s respectively. Furthermore the integrator was added in order to get motor position instead of motor angular velocity because it is the position that is going to be controlled.

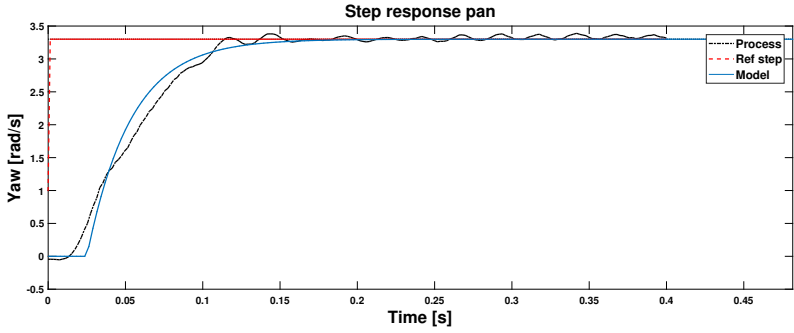


Figure 6.4 Experimental and simulated step response of the pan-system

In Figure 6.5 it can also be seen that the tilt-system is described by the same model. This was not verified though, since the experiment was hard to perform on the tilt-system because of the tilt limitation described in Equation 3.2. However, it seemed like a reasonable assumption due to the similarities in mechanical setup and in configuration of the motion profile in the motion controllers.

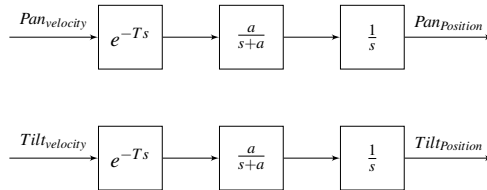


Figure 6.5 Model of the system

Note that the models described above are valid only for a certain motion profile configuration. This is because the motion profile is a trajectory generator for the stepper motor. A change in configuration is thereby the same thing as a change in dynamics. In Figure 6.6 the step responses of three different configurations are illustrated. It is clear that all three responses cannot be described by the same dynamical model. However, the configuration shown in Figure 6.4 is the standard configuration which is used in general. Due to that the dynamical models in Figure 6.5 can be used for control design.

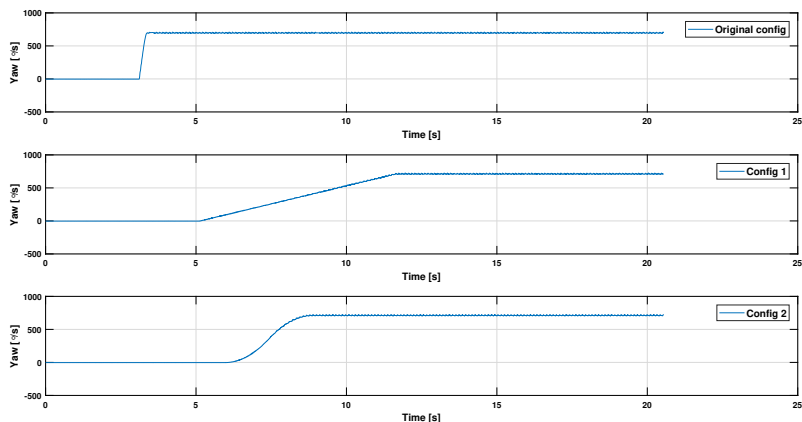


Figure 6.6 Step response (pan) with three different motion profile configurations

7

Trajectory generation

In robotics and motor control it is often required to generate a trajectory in order to reach a desired velocity or position. How the trajectory looks is often set by constraints on the process or the actuators. Furthermore, trajectory generation can be used in both open and closed loop systems as Figure 7.1 shows.

The PTZ-camera model Q6155-E uses two motion controllers that act as independent velocity trajectory generators for the pan and tilt stepper motors, respectively. Their location in the system architecture can be seen in Figure 3.2. This means that the motion controllers decide how the pan and tilt systems are allowed to move. In turn, it means that constraints are put on the physical models described in Chapter 6. Because of this, the behavior of the pan and tilt systems directly depend on how the trajectories are generated, which is the topic of this chapter. Some details on the calculations of phase durations are collected in Appendix A. The generated profile could furthermore be used in model predictive control, but that is outside the scope of this thesis.

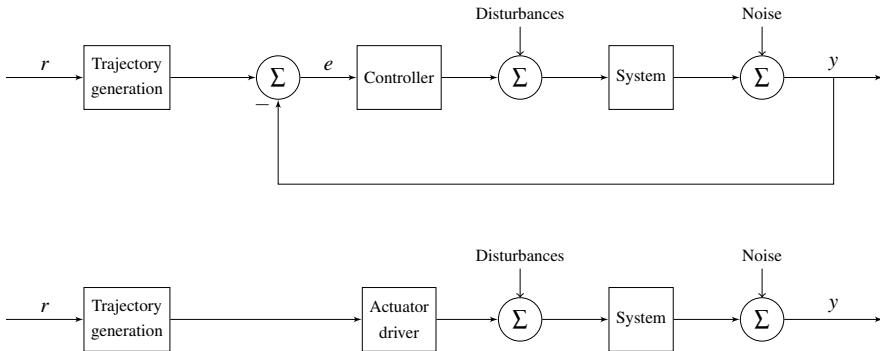


Figure 7.1 Closed (top) and open (bottom) loop system

7.1 Motion profiles

Two common motion profiles are the S-profile and the trapezoidal profile. Other names are third order trajectory profile and second order trajectory profile. Figure 7.2 shows a comparison between the two. The order depends on where you want to limit your system. The trapezoidal profile limits acceleration, and the S-profile limits the time derivative of acceleration, often referred to as jerk. This means that the order of the profile polynomial increases when n in Expression 7.1 increases.

$$\frac{d^n x(t)}{dt^n} \quad n \in \mathbb{N} \quad n \geq 2 \quad (7.1)$$

The limitation changes in a step-formed manner between time intervals, as can be seen in the acceleration limitation for the trapezoidal profile in Figure 7.2. When the order increases more time intervals (phases) will be present in the limitation graph. The disadvantage of higher order trajectory profiles is increased complexity and this will result in an increased trajectory execution time. However, this is usually compensated for, due to reduced settling time. One of the advantages is that higher order trajectory profiles inherently have a lower energy content at higher frequencies. This means that resonances at higher frequency ranges will not excite the process very much. Other advantages are less demands on actuators and processes, compared to step-formed references. This will result in less wear and tear.

More about this topic can be found in for instance. [Linderroth, 2013] [Lewin, 2007] [Haschke et al., 2008] [Lambrechts et al., 2005] [Kröger et al., 2006] [Reflexxes, 2017] [Kröger and Wahl, 2010]

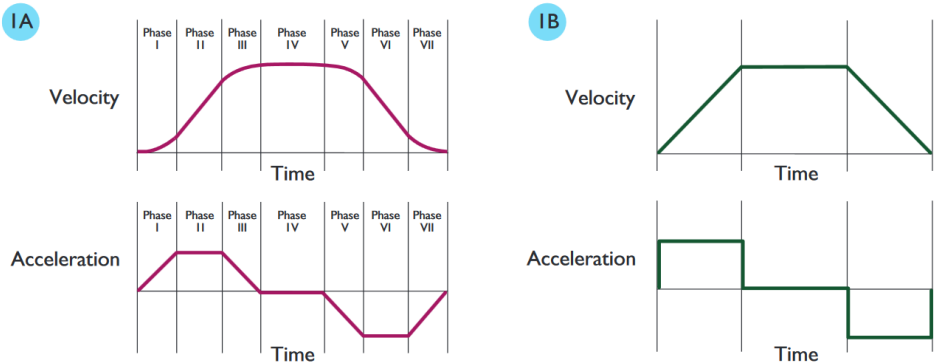


Figure 7.2 (1A) S-curve and (1B) trapezoidal profile [Lewin, 2007]

S-profile

To reach a target it is often desired to have a "smooth" transition. This is done by limiting the derivative of acceleration with respect to time, also called jerk (j). This motion is often called an S-profile or S-curve in motor controller terms. The S-profile can often be used in two different modes, velocity mode or position mode. In velocity mode the goal is to reach a constant velocity whereas the goal of the position mode is to reach a certain position. The mode that is used decides the number of constraints and phases that are needed. Figures 7.2 and 7.3 show the S-profile in position mode. There are seven phases (i) that correspond to seven different time intervals in Equation 7.2, and there are seven steps between the phases in the jerk-profile, as seen in Equation 7.3.

$$\Delta t_i \quad i = \{1, 2, 3 \dots 7\} \quad (7.2)$$

$$\vec{j} = [\pm J_1, 0, -J_3, 0, -J_5, 0, J_7] \quad (7.3)$$

The double sign for jerk J_1 of the first phase shall indicate that J_1 switches sign, if the initial acceleration exceeds its limit, resulting in a double deceleration profile within phases 1–3. Equations 7.4–7.6 show the set of polynomial equations for acceleration, velocity and position, which have to be solved for obtaining the phase durations Δt_i .

$$a_i = a_{i-1} + j_i \Delta t_i \quad i = \{1, 2, 3 \dots 7\} \quad (7.4)$$

$$v_i = v_{i-1} + a_{i-1} \Delta t_i + \frac{1}{2} j_i \Delta t_i^2 \quad i = \{1, 2, 3 \dots 7\} \quad (7.5)$$

$$x_i = x_{i-1} + v_{i-1} \Delta t_i + \frac{1}{2} a_{i-1} \Delta t_i^2 + \frac{1}{6} j_i \Delta t_i^3 \quad i = \{1, 2, 3 \dots 7\} \quad (7.6)$$

Note that the left side of Equations 7.4–7.6 are the end values for each phase. In position mode there are constraints on acceleration and velocity in some of the phases, namely:

$$a_3 = 0 \quad a_4 = 0 \quad a_7 = 0 \quad v_7 = 0 \quad (7.7)$$

There are also limitations in maximum acceleration, maximum deceleration and maximum velocity. This will set constraints on phases 2, 4 and 6:

$$a_2 = \pm a_{max_{acc}} \quad a_6 = \mp a_{max_{dec}} \quad v_4 = \pm v_{max} \quad (7.8)$$

Note, that in order to calculate negative position the negative of \vec{j} and the bottom signs of Equation 7.8 can be used.

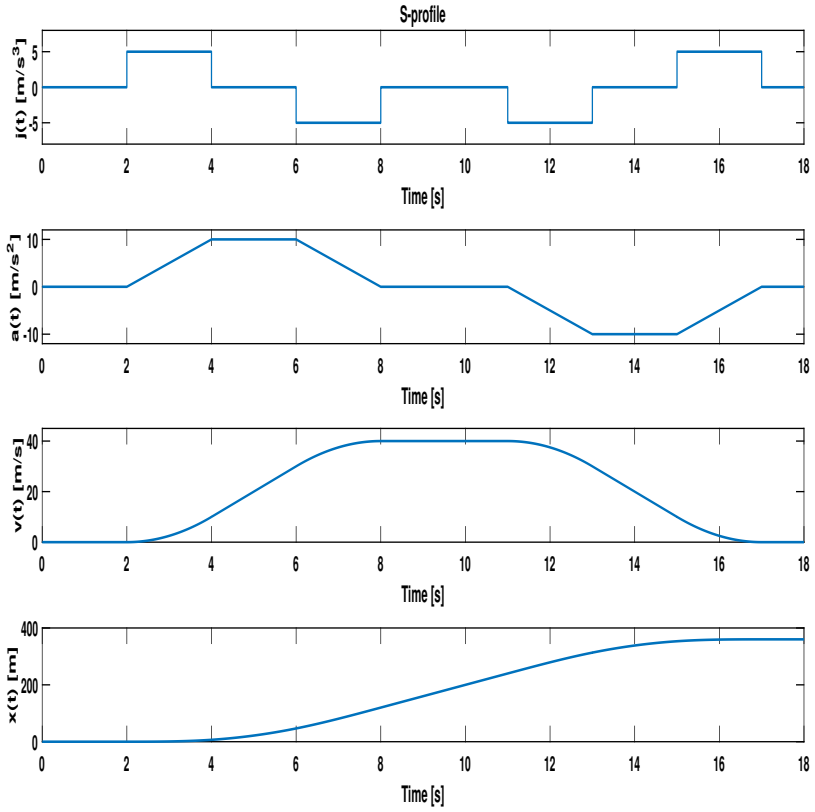


Figure 7.3 An example of a S-profile in position mode

In velocity mode only three phases are needed in order to reach the target velocity. This means that there are just three phase durations and jerk-profile steps, as seen in Equations 7.9–7.10. Moreover, Equations 7.4–7.6 can be used to calculate acceleration, velocity and position for the three phases. Equation 7.11 shows the constraints and limits that are needed.

$$\Delta t_i \quad i = \{1, 2, 3\} \quad (7.9)$$

$$\vec{j} = [\pm J_1, 0, -J_3] \quad (7.10)$$

$$a_2 = \pm a_{max_{acc}} \quad a_3 = 0 \quad (7.11)$$

7.2 Motion controllers

Axis uses the motion controller chip TMC4361A-LA from Trinamic and it is used for controlling the stepper motors [Trinamic, 2017]. The chip has capability to generate both S-profiles and trapezoidal profiles. The current configuration is S-profile motion and both the pan and tilt system are equipped with a motion controller in an open loop configuration as Figure 7.4 shows. However, the encoders are used to correct the position after a movement has been carried out. Moreover, the motion controllers can be used in both velocity and position mode.

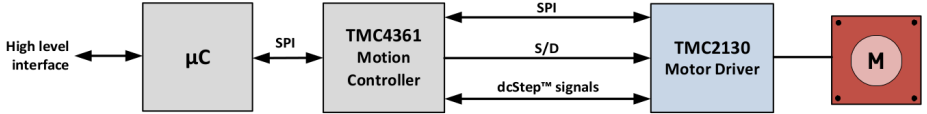


Figure 7.4 An open loop case with TMC4361A-LA [Trinamic, 2017]

In this thesis the motion controllers have been used in velocity mode for both the pan and tilt system. The jerk profile was not changed because it was already optimized for Axis' needs. This means that all phases in the jerk-profile are different, as seen in Equation 7.12.

$$J_1 \neq J_3 \neq J_5 \neq J_7 \quad (7.12)$$

Depending on if $J_3 = J_7$ or $J_3 \neq J_7$ the behavior motion controllers will be different in the case of a u-turn¹. If $J_3 \neq J_7$ the behavior will be as shown in Figure 7.5 and in the case $J_3 = J_7$ the behavior will be as seen in Figure 7.6. It is clear that $J_3 = J_7$ creates the fastest possible u-turn¹. This is because the acceleration does not have to be 0 when the velocity is 0. Furthermore, $J_3 = J_7$ can be described by Equations 7.9–7.11. For the case $J_3 \neq J_7$ it gets a little bit trickier, but this can be solved with Equations 7.9–7.11 as well, by introducing some logics. This can be done as follows:

First set the target velocity to 0 with the following constraints:

$$\vec{j}_{acc} = [\pm J_5, 0, -J_7] \quad a_2 = \pm a_{max_{dec}} \quad a_3 = 0 \quad (7.13)$$

Then use the end value, i.e., acceleration, velocity and position, as initial values and also set the "real" target velocity with following constraints:

$$\vec{j}_{acc} = [\pm J_1, 0, -J_3] \quad a_2 = \pm a_{max_{acc}} \quad a_3 = 0 \quad (7.14)$$

¹ A u-turn means that the sign and possibly the absolute value of the current velocity changes.

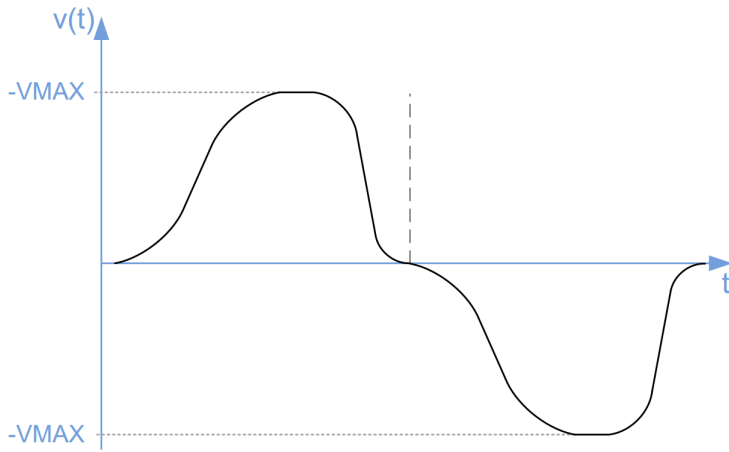


Figure 7.5 U-turn when $J_3 \neq J_7$. The dashed line indicates where the velocity and acceleration are equal to 0 [Trinamic, 2017]

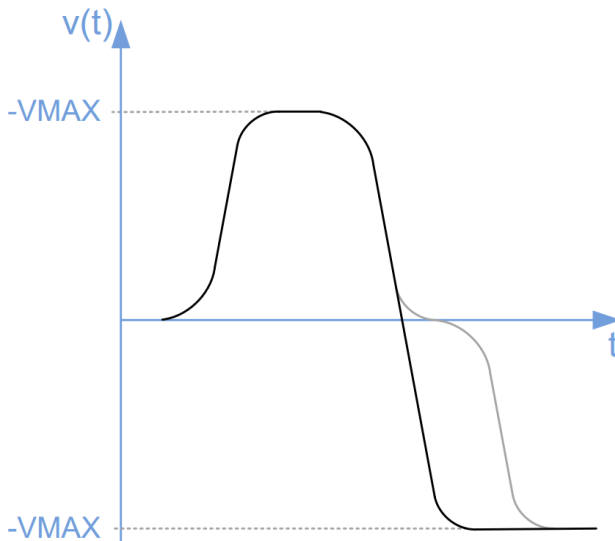


Figure 7.6 U-turn when $J_3 = J_7$. The gray line shows the case $J_3 \neq J_7$ [Trinamic, 2017]

7.3 Discussion

When using a desired motion profile like the S-curve for the reference the problem is to calculate the Δt_i , representing how long each phase is, based on the limits for acceleration, velocity and jerk. Moreover, the initial values are known, i.e., a_0 , v_0 and x_0 . The mode that is used will affect the complexity, i.e., the position mode results in more phases, limits and constraints compared to the velocity mode. As mentioned earlier, the velocity mode was used and in Appendix A a more detailed calculation of Δt_i is provided. The motion controllers add additional complexity depending on which of the cases $J_3 = J_7$ or $J_3 \neq J_7$ that is used. For the latter case, logics can be added to the calculations done in Appendix A. If the motion controller behavior would have been used in the control design, a better controller performance had most certainly been obtained.

8

Control and implementation

The control design of the thesis was done in the MATLAB extensions Control System Designer and Simulink with the pan and tilt models described in Chapter 6. Due to the assumption that the systems did not affect each other through cross-couplings and their identical models only one controller had to be designed. The choice of controller fell on the PID-controller which is a relatively simple but often very effective controller. It is typically used as shown in Figure 8.1, where the three left blocks represents the controller. However, some modification of the algorithm was needed to get a successful implementation.

The upcoming sections will describe general control theory, the design process and the implementation of the whole control system, including orientation estimation and reference generation. Moreover the resulting controller will be presented in a simulation.

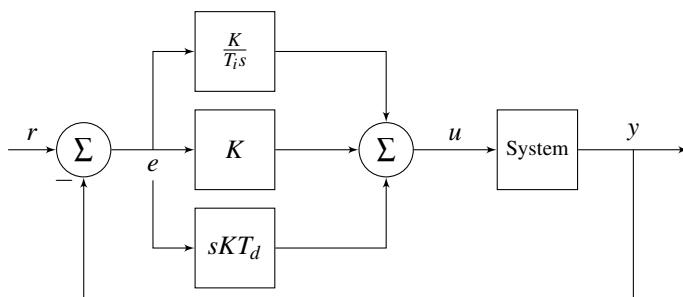


Figure 8.1 A typical PID control setup

8.1 PID-control

PID-control is a frequently used method for controlling dynamical systems. Much for its simplicity and intuitive behaviour. In its most simple form it can be described by the equation

$$u(t) = K \cdot \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (8.1)$$

where e is the control error, T_i the integral time, T_d the derivative time and K the proportional gain. The name PID refers to the three terms in Equation 8.1, proportional, integral and derivative [Wittenmark et al., 2014]

The proportional term of the controller gives a contribution to the control signal that is proportional to the control error. This means that a higher value of K will result in a greater control signal and therefore in most cases also a faster process response. However, the stability of the closed loop system will decrease with an increasing value of K . Another issue with the proportional part is that it can give a steady state error [Thomas, 2008] [Åström and Murray, 2008].

The second term in Equation 8.1 is the integral term. The integral time T_i is a design parameter which decides the level of contribution that the integral term will have on the control signal. The advantage with this term is that it can take care of the steady state error which the proportional part cannot handle and compensate for the stationary effect of load disturbances. An intuitive way to realize this is to think of the integral term as an accumulation of past errors, which will not stop accumulate until the error is zero. However, a low value of T_i can cause a very fast accumulation and can therefore give overshoots in the process output. Moreover the accumulation can in the worst scenario cause the actuator to saturate. In this case the feedback will stop working because the process output can change without a reaction from the actuator. This is known as integrator windup and can be avoided with different kinds of anti-windup methods [Åström and Murray, 2008].

The third and last term in Equation 8.1 is the derivative term which contains the design parameter T_d . The term provides two properties. The first is an increasing speed of the control loop and the second is improved stability. Increasing speed can be realized in the case of an increasing error. In this situation the derivative part will give the control signal an extra contribution meaning that the error can be reduced faster. Improved stability can be realized in the opposite situation, namely when the error is decreasing. In this case the derivative term will decrease the control signal and thus giving a smaller overshoot. In other words the derivative term can be described as a predictor [Thomas, 2008] [Årzén, 2014].

Anti-windup

One method to solve the problem with integrator windup is a method known as tracking. The block diagram in Figure 8.2 describes the PID-controller with tracking included. The idea of the method is to provide an extra feedback path that measures the difference, e_s , between the calculated and actual control signal and then feeds it to the integrator. This will then force the integrator output towards a value that makes the integrator input equal to zero. This will however, only happen when there is an actuator saturation, i.e. $u_1 \neq u_2$. Moreover the gain $1/T_t$ is called the tracking time constant and decides how fast the integrator output will converge towards the value that makes the input zero [Årzén, 2014].

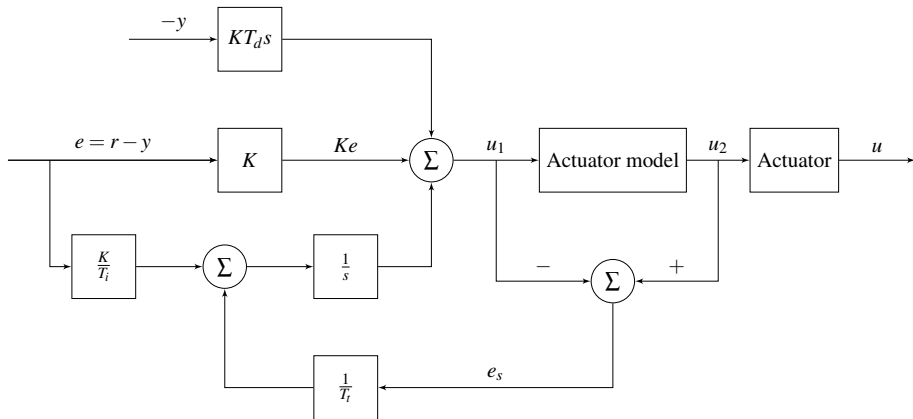


Figure 8.2 Anti-windup

Derivative filtering

In practical implementations of the PID-controller the derivative term is usually low-pass filtered because of high frequency measurement noise. In this way large amplification of high frequencies is avoided, resulting in better control and less wear on actuators [Årzén, 2014] [Thomas, 2008]. The derivative term is thus approximated by

$$sT_d \approx \frac{sT_d}{1 + \frac{sT_d}{N}} \quad (8.2)$$

The approximation holds for low frequencies with the design parameter N setting the limit, as it decides the maximum gain of the filter. Figure 8.3 illustrates the approximation in a Bode plot.

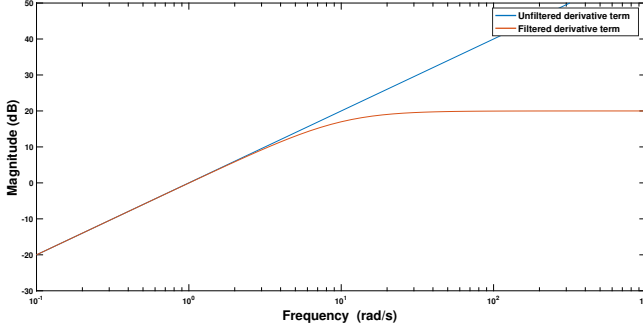


Figure 8.3 Illustrates the approximation of the derivative term with $N = 10$ and $T_d = 1$

Set-point weighting

In the original PID-controller, presented in Equation 8.1, the set point and the measured value are treated in the same way. However, this is not the way the PID-controller usually is implemented in practice, due to empirically results showing that treating them differently gives improved performance. Treating them differently does in this case mean that only a fraction of the set point value is used in the proportional and derivative term. For the derivative term this means that step changes in the set point value will not saturate the control signal [Årzén, 2014]. The modified version of the PID-controller in the frequency domain, including derivative filtering and set-point weighting, is given by

$$U(s) = K \cdot \left[\beta R(s) - Y(s) + \frac{1}{sT_i} (R(s) - Y(s)) + \frac{sT_d}{1 + \frac{sT_d}{N}} (\gamma R(s) - Y(s)) \right] \quad (8.3)$$

where β and γ are the set-point weights, $R(s)$ the set point and $Y(s)$ the measured output [Årzén, 2014].

Discrete PID-control

The PID-controller described above is expressed in continuous time and can thereby not be implemented in the camera software. To implement it a discretization of each of the three terms is needed. The proportional part however, can be implemented just as it is. It is the derivative and integral terms that need to be approximated. This can be done by using backward and forward Euler approximation, which in the frequency domain are described as:

$$s \approx \frac{z-1}{zh} \quad (8.4)$$

$$s \approx \frac{z-1}{h} \quad (8.5)$$

where Equation 8.4 is backward Euler, 8.5 forward Euler and h the sampling interval of the control algorithm [Wittenmark et al., 2014]. If the derivative term is approximated with backward difference and the integral term with forward difference the control law described in Equation 8.3 will result in:

$$U(z) = K \cdot \left[\beta R(z) - Y(z) + \frac{h}{(z-1)T_i} (R(z) - Y(z)) + \frac{\frac{z-1}{zh} T_d}{1 + \frac{\frac{z-1}{zh} T_d}{N}} (\gamma R(z) - Y(z)) \right] \quad (8.6)$$

which is the control law that was finally implemented in camera software, but of course, converted to the sampled time domain first.

8.2 Control structures

Instead of using the PID-controller as shown in Figure 8.1, it can be put in a control structure, meaning that the controller then will operate together with other controllers. The structures can vary depending on the type of system that is to be controlled and the set of controller requirements. It can be a system where disturbance compensation is of most importance or a system where actuators have different limits and by combining them a better control performance is achieved. Some examples of controller structures are cascade control, mid-ranging control and internal model control (IMC).

Two controller structures that was of interest to investigate in this thesis was:

- Cascade control
- Mid-ranging control

Cascade control

Cascade control is a control method that is commonly used when the process that is to be controlled can be split into two or more separate processes in series. In the case of two processes two control loops can be constructed, see Figure 8.4. The

main advantage with this structure is that the disturbance to left can be compensated for before it reaches the output y [Thomas, 2008].

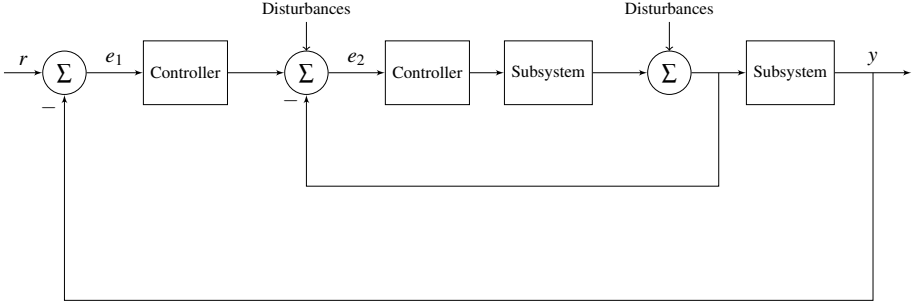


Figure 8.4 Cascade control

In this thesis cascade control was relevant since there was a possibility to create a feedback path of the pan and tilt angular velocity. Either by converting gyroscope rates to pan and tilt rates or by taking the derivative of the encoder signals. This control structure was, however, discarded due to the fact that disturbances do not primarily enter the system in this way. More about this in Section 8.3.

Mid-ranging control

Mid-ranging control is a control method that can be used when two actuators can be used for controlling the same control variable but one of them is easier to saturate than the other one. The concept is shown in Figure 8.5. The idea is to let the controller C_2 control the actuator which can affect the control variable more, to make it possible for the controller C_1 to work in the middle of its control interval. By doing this C_1 will not saturate equally easy when trying to track the reference signal r_1 [Forsman, 2005].

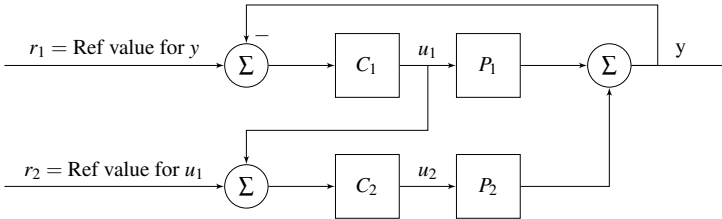


Figure 8.5 Mid-ranging control

In this thesis Mid-ranging was considered since the camera already uses electronic image stabilization which cannot compensate for disturbances with large am-

plitude. This can, however, be done more successfully with the mechanical stabilization. A Mid-ranging solution then seemed reasonable to give the best stabilization possible. Unfortunately this was never tested due to lack of time. The primary objective was to get the mechanical stabilization to work.

8.3 Control design

The control design was done in continuous time and then discretized to get the final controller. This approach was taken mainly because of greater experience in continuous than discrete control design.

To study the controller performance the feedback loop in Figure 8.6 was considered, where C represents the PID-controller, P the pan or tilt process described in Chapter 6, F a feedforward which in this case was set to one, d a load disturbance and n encoder measurement noise. Note that the load disturbance is a disturbance that acts directly on the pan and tilt and not on the whole camera body.

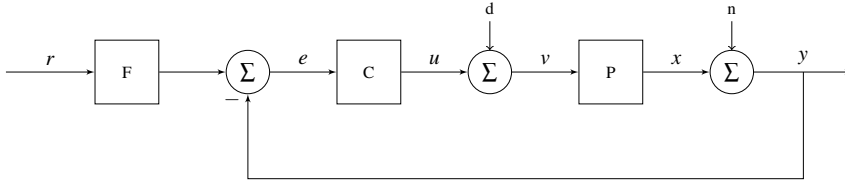


Figure 8.6 General feedback system

To obtain the controller parameters frequency domain design was used. The idea of the method is to change the open loop gain, i.e. $L = PC$, to obtain the desired closed loop properties [Åström and Murray, 2008]. However, there are more signal relationships than just the relation between r and y that can be investigated. The influence from all external signals on the internal signals in Figure 8.6 is given by:

$$X = \frac{P}{1+PC}D - \frac{PC}{1+PC}N + \frac{PCF}{1+PC}R \quad (8.7)$$

$$Y = \frac{P}{1+PC}D + \frac{1}{1+PC}N + \frac{PCF}{1+PC}R \quad (8.8)$$

$$U = -\frac{PC}{1+PC}D - \frac{C}{1+PC}N + \frac{CF}{1+PC}R \quad (8.9)$$

$$E = -\frac{P}{1+PC}D - \frac{1}{1+PC}N + \frac{F}{1+PC}R \quad (8.10)$$

$$V = \frac{1}{1+PC}D - \frac{C}{1+PC}N + \frac{CF}{1+PC}R \quad (8.11)$$

where the uppercase notation of the signals indicate frequency domain representation. As mentioned above $F = 1$, meaning that the transfer functions reduce to:

$$\begin{aligned} S &= \frac{1}{1+PC} & PS &= \frac{P}{1+PC} \\ T &= \frac{PC}{1+PC} & CS &= \frac{C}{1+PC} \end{aligned} \quad (8.12)$$

which are known as the *Gang of Four* [Åström and Murray, 2008].

CS and PS were not investigated in particular because of little encoder noise, n , and due to the fact that disturbances mostly do not enter the system where the load disturbance, d , enters. Disturbances are rather handled by the reference generator which creates references that suppresses the disturbances if reference tracking works properly. This means that the most important transfer to study was the the Complementary sensitivity function, T , since it describes the relation between r and y . S was also studied to look at the relationship between the signals r and e .

The controller was designed in Control System Designer which is a tool that can be used to graphically modify the open loop gain, PC , and simultaneously see the effect on the closed loop Bode plot and step response. However, a control signal saturation cannot be set. Because of this the controller was also simulated in Simulink. The discrete version of the obtained analog controller was also simulated.

Simulation result and discussion

In the control simulation three tests were made on the closed loop system, namely:

- Step response analysis
- Sinusoidal response analysis
- Ramp response analysis

All the tests were carried out with a discrete PID-controller with a sampling interval of 50ms.

The step response result is shown Figure 8.7. As can be seen there is a small overshoot, however, this was accepted due to reduction of rise time. Moreover it can be seen that the error converges towards zero but not very quickly. The error is however small enough after approximately 0.7s.

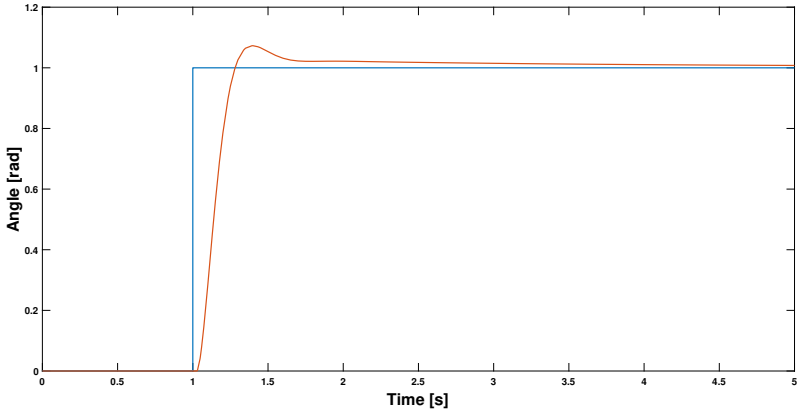


Figure 8.7 Simulated closed loop step response

The sinusoidal response was studied for two frequencies, 1 and 1.5Hz. Figures 8.8 and 8.9 show the response them both. The 1Hz response has no reduction of amplitude, but there is a time delay present due to the time delay in the dynamical model. This made it impossible to get perfect tracking.

The 1.5Hz response shows a significant decrease in amplitude. This means that the reference signal has higher frequency than the closed-loop bandwidth. However, the requirement of the closed-loop bandwidth was set to 1Hz, meaning that the controller worked properly.

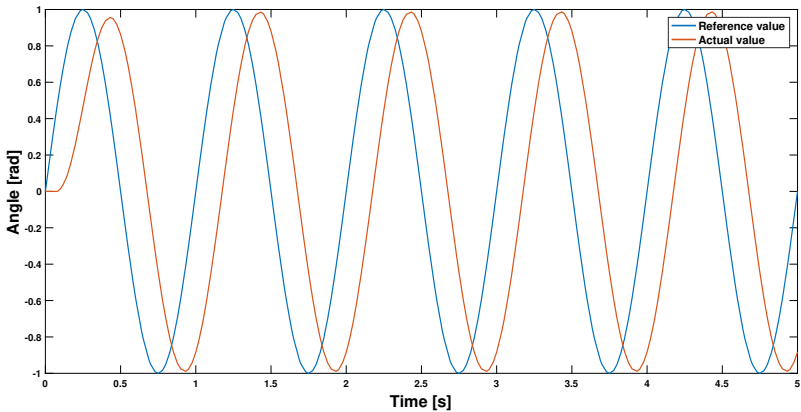


Figure 8.8 Simulated closed loop response with a 1Hz sinusoidal reference signal

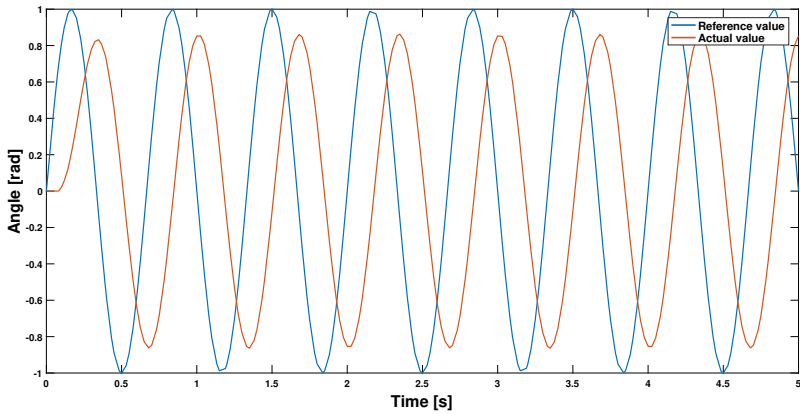


Figure 8.9 Simulated closed loop response with a 1.5Hz sinusoidal reference signal

Figures 8.10 and 8.11 show the ramp response and how the error changes over time. It is clear that the error does not reach zero in 1.5s but Figure 8.11 shows that it is decreasing. In Figure 8.12 the error is studied over a longer time and it can be seen that it converges towards zero. However, the convergence time is too long for short alternating ramp references as seen in Figures 8.13 and 8.14. The error convergence can be made shorter though, if some of the controller parameters are changed. This can be seen in Figure 8.15. The controller parameters were however not used on the real process since the ramp response behavior was discovered during real experiments. The ramp simulations were made afterwards, in order to solve the problem.

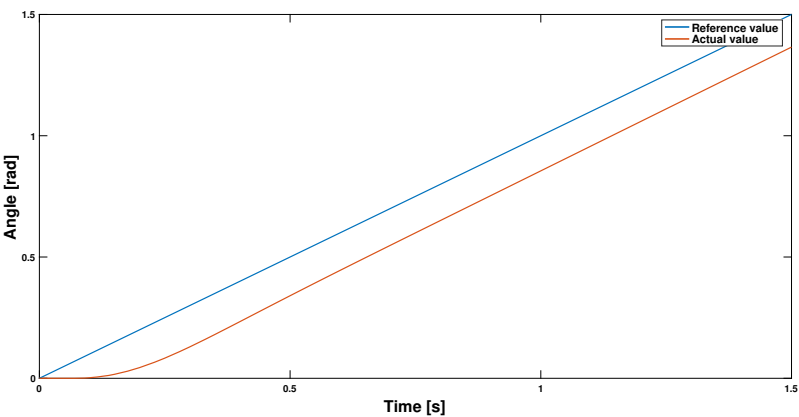


Figure 8.10 Simulated closed loop ramp response

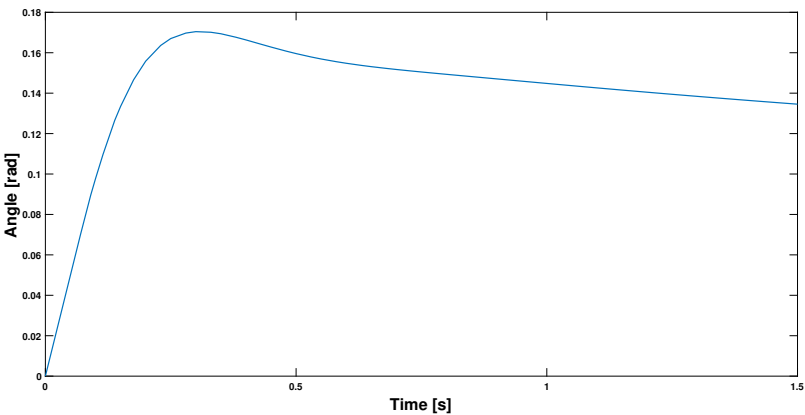


Figure 8.11 Error over 1.5s with a ramp reference signal

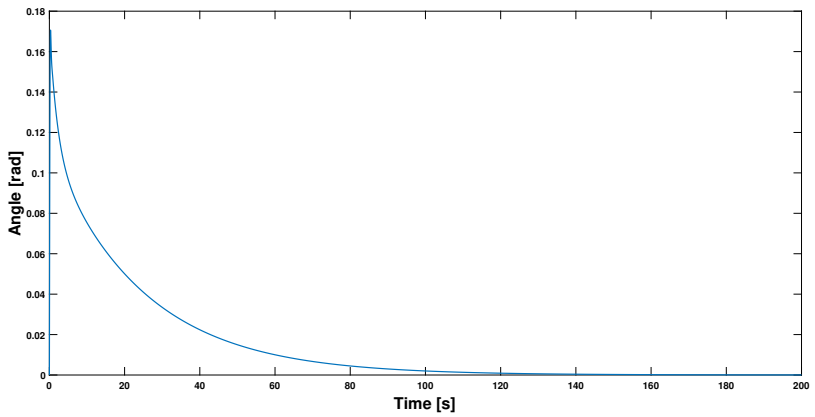


Figure 8.12 Error over 200s with a ramp reference signal

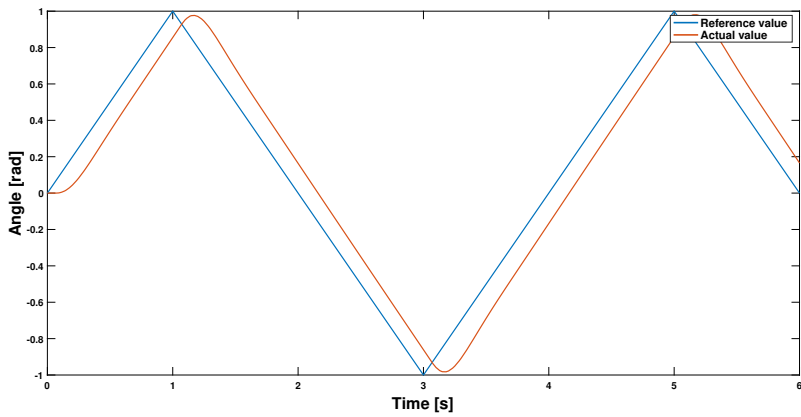


Figure 8.13 Simulated closed loop response with a triangle wave reference signal

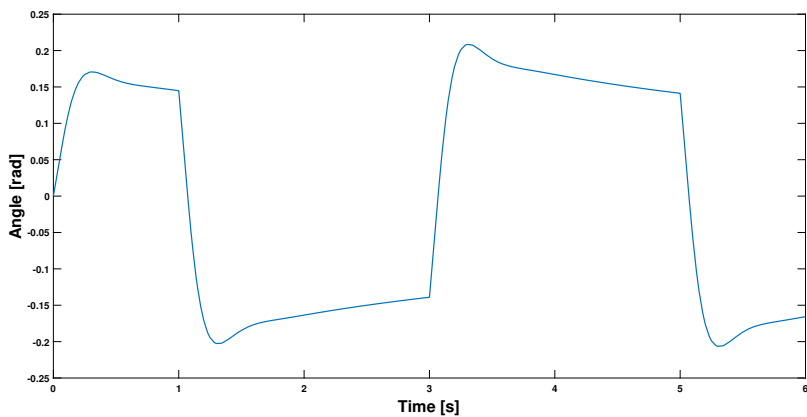


Figure 8.14 Error with a triangle wave reference signal

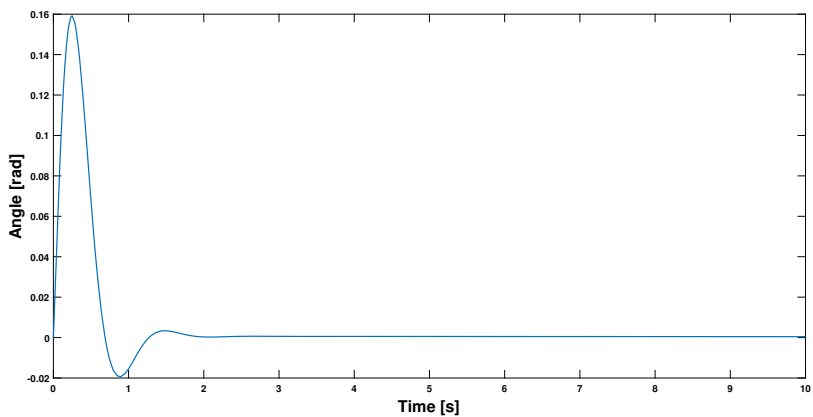


Figure 8.15 Error with a ramp reference signal (changed controller parameters)

In order to explain the difference in convergence in Figures 8.12 and 8.15 the final value theorem can be used [Thomas, 2008]. First consider the transfer function which describes the relationship between the reference and the error, which looks as follows:

$$E(s) = \frac{1}{1+PC}R(s) \quad (8.13)$$

where the open loop transfer function is described by:

$$PC = e^{-Ts} \frac{a}{s+a} \frac{1}{s} \left(K + \frac{1}{s} \frac{K}{T_i} + sKT_d \right) \quad (8.14)$$

If the reference signal is a ramp it can be described by:

$$R = \frac{1}{s^2} \quad (8.15)$$

These three equations make it possible to use the final value theorem in order to investigate the error convergence. The final value theorem states the following:

$$\lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) \iff \quad (8.16)$$

$$\lim_{s \rightarrow 0} \frac{s}{s^2 + e^{-Ts} \frac{a}{s+a} \left(sK + \frac{K}{T_i} + s^2KT_d \right)} = 0 \quad (8.17)$$

This means that the error will converge to zero in both cases. However, if T_i is small the convergence will be faster. This is also what the simulation shows. Note that the final value theorem can not be used for the sinusoidal reference as the closed-loop system final value does not exist in this case.

8.4 Hardware implementation

To get a final result of how the camera actually managed to compensate for disturbances, an implementation had to be done. However, it was first needed to study existing code, in order to integrate the new in the best way possible. After the analysis the implementation was done on the camera MCPU and everything was written in C-code.

Code structure

In Figure 8.17 a block diagram of the whole system structure is shown. It contains all parts described in the thesis. Furthermore a flowchart representing the algorithm that was implemented can be seen in Figure 8.16. The implemented program contains the following parts:

Initialization, Setup data cache communication and Setup System

In the setup phase the program is first initialized and then the data cache communication is setup. The data cache is an API that handles many different data structures in the camera. Two examples are the IMU and encoder data. So with the data cache setup it is possible to get the raw sensor data that is needed for the control system.

In the setup system block the parameters of the controller, Kalman and Complementary filter are set. The user sets the parameters in the configure file seen in Figure 8.16.

Measurements and orientation estimation

Measurements and orientation estimation is the first part of the main loop in the program structure, which is set to run at 20Hz. Here the IMU data is read from the data cache in chunks due to the fact that the IMU data gets updated at a frequency of 500Hz. Moreover the Kalman or Complementary filter executes their algorithms and the yaw rate is numerically integrated, as described in Chapter 4. The size of the chunk decides how many estimation iterations that needs to be done in order to obtain the latest Euler angles that is needed for the reference generation.

The encoder data is also read in this part, but it is updated at a frequency of 50Hz. However only the latest measurement is used for the reference generation.

Reference generation

The second part of the main loop is the reference generation. The generation begins by creating the rotation matrix described in Equation 5.1 and then Euler angle extraction is done as described in Chapter 5. Two of the extracted angles are then forwarded as references for the PID-controllers.

PID-control

The PID-control is the final part of the main loop and it is implemented with the time domain representation of Equation 8.6. If $\gamma = 0$ the conversion of the three terms can be expressed as follows [Wittenmark et al., 2014]:

$$P(k) = K(\beta r(k) - y(k)) \quad (8.18)$$

$$I(k+1) = I(k) + \frac{Kh}{T_i}(r(k) - y(k)) + \frac{h}{T_i}(u_2(k) - u_1(k)) \quad (8.19)$$

$$D(k) = \frac{T_d}{T_d + Nh}D(k-1) - \frac{KT_dN}{T_d + Nh}(y(k) - y(k-1)) \quad (8.20)$$

where the third term in Equation 8.19 adds tracking to the integral part, see Figure 8.2. Moreover, the structure of the code looks as follows:

$$u_1(k) = P(k) + I(k) + D(k) \quad (8.21)$$

$$u_2(k) = \text{sat}(u_1, \text{upperLimit}, \text{lowerLimit}) \quad (8.22)$$

$$I(k+1) = I(k) + \frac{Kh}{T_i}(r(k) - y(k)) + \frac{h}{T_i}(u_2(k) - u_1(k)) \quad (8.23)$$

where *sat* is a function that checks if the calculated control signal is within its limits. If not within, the control signal is set to either the upper or lower limit of the actuator, depending on its sign [Árzén, 2014]. In this case the limits are $700^\circ/s$ for pan and $500^\circ/s$ for tilt.

System standby

After the PID-execution a wait function is implemented. This function sets the system in standby mode until a certain time is reached. In this way the desired main loop sampling frequency of 20Hz is obtained.

To keep track of time a time check is done just before the main loop is started and in the end of the main loop the sampling time is added to that time. By doing this in every main loop iteration the time will move forward and thereby it is possible to use the wait function to set the system in standby until the time that is obtained from the addition. However, if the system is supposed to sleep until a time that already lies in the past, which might happen, a new time check will be made inside the wait function and the addition will then start from there [Árzén, 2014].

Stop and Clear system

When the system is running a function that listens to certain user inputs is also present. In this case this function is used for shutting down the system, meaning that the main loop breaks, the data cache communication is closed and memory is deallocated.

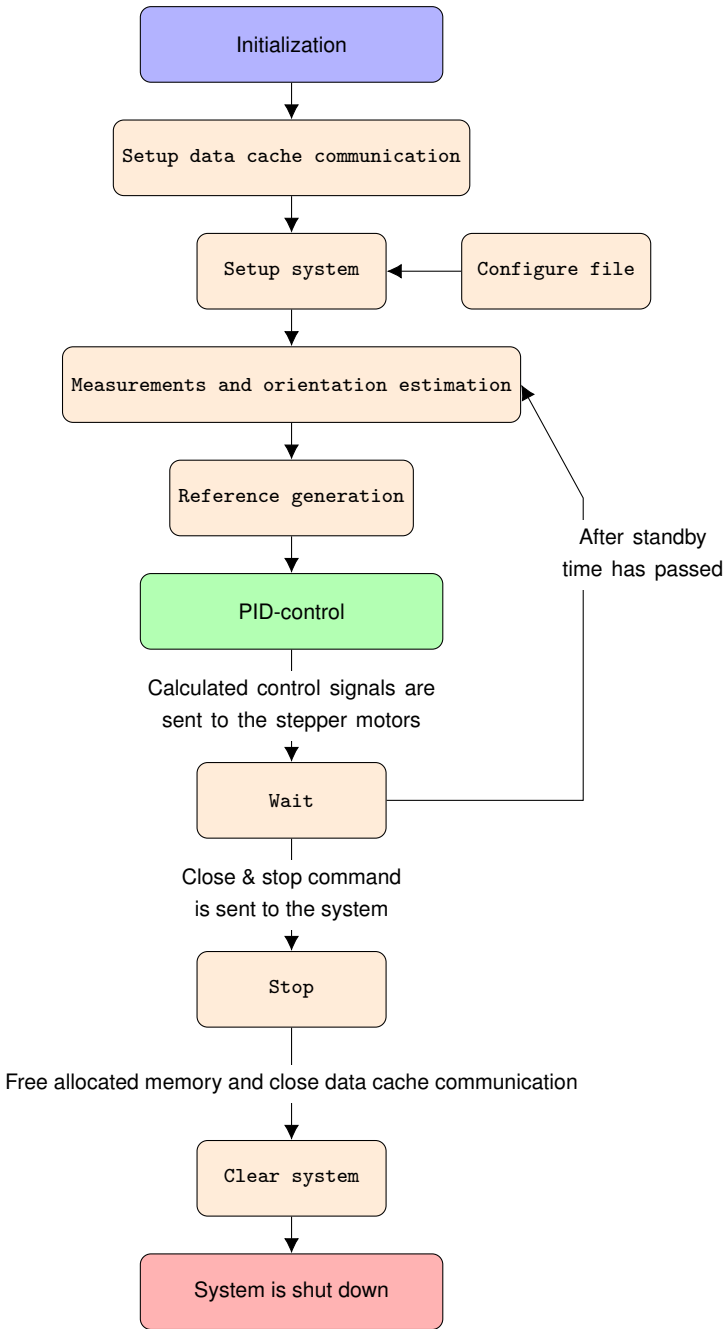


Figure 8.16 Flowchart representing the final algorithm

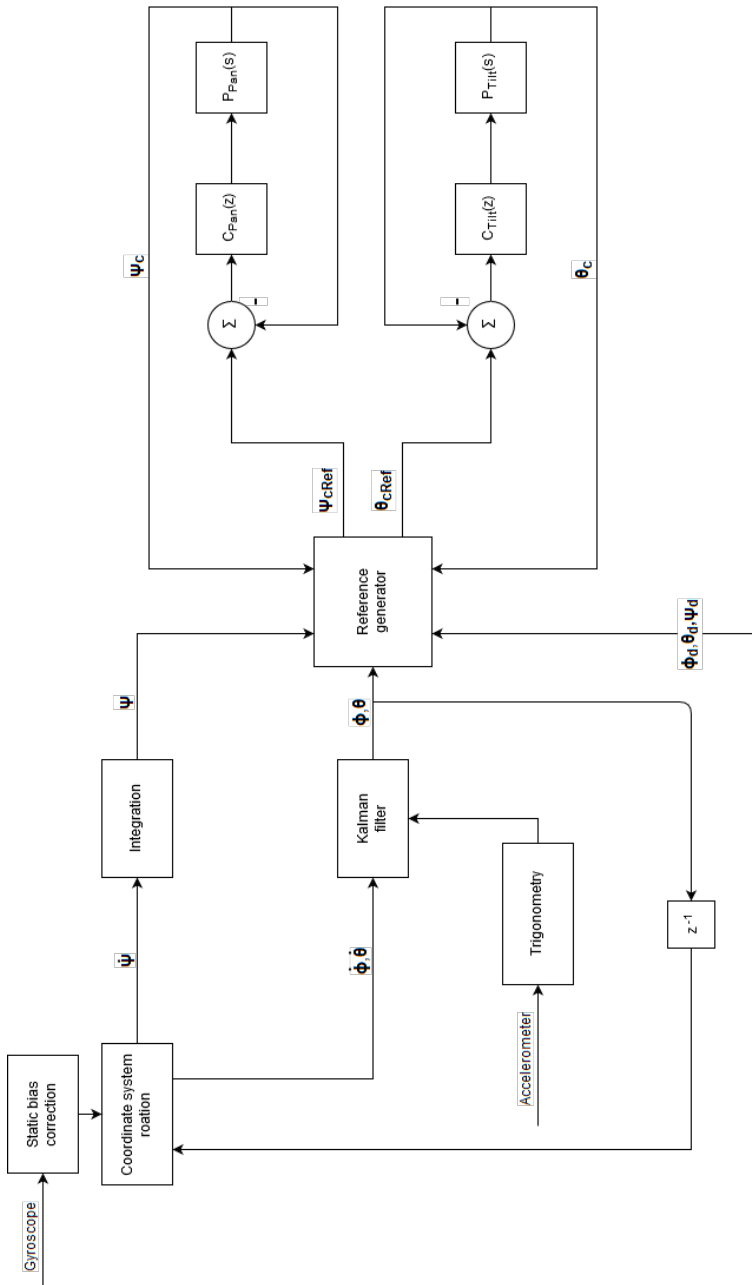


Figure 8.17 Final system structure

9

Results

The implementation made it possible to carry out experiments on the real process/camera and compare it with the simulation results. However, the experiments were carried out for both pan and tilt, which was not necessary in the simulation. The following experiments were done:

- Closed loop step response
- Closed loop sinusoidal response
- Shake experiments

9.1 Closed loop step response

The results of the step response experiments can be seen in Figures 9.1 - 9.3. Note that the simulated response is also present.

Both the pan and tilt experiment differ in overshoot compared to the simulation, see Figures 9.1 and 9.2. However, the model seems to describe the system good enough for these cases.

In Figure 9.3 though, it can be seen that the tilt overshoot varies depending on step magnitude. This indicates that there is a nonlinearity present in the tilt system. Hence a linear time invariant model might not be good enough in all cases. This was however never investigated in depth due to lack of time.

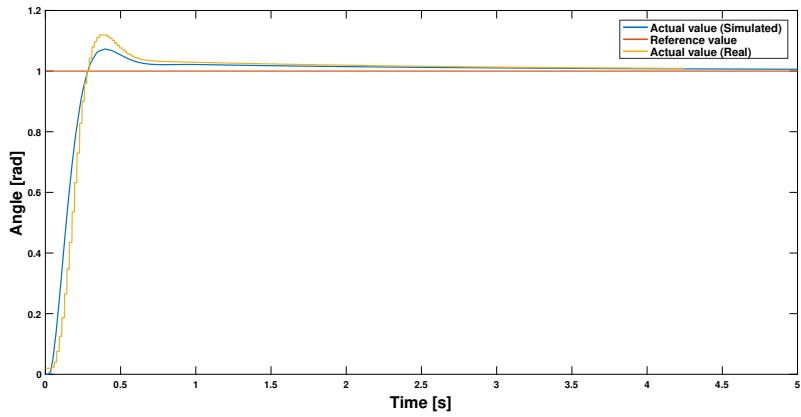


Figure 9.1 Real and simulated closed loop step response (pan)

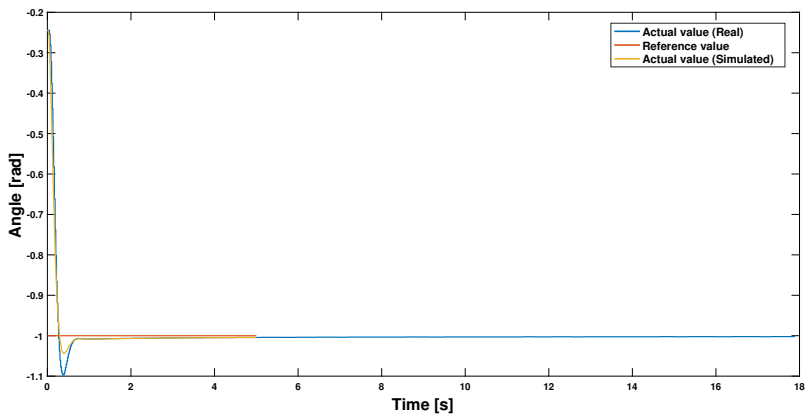


Figure 9.2 Real and simulated closed loop step response (tilt)

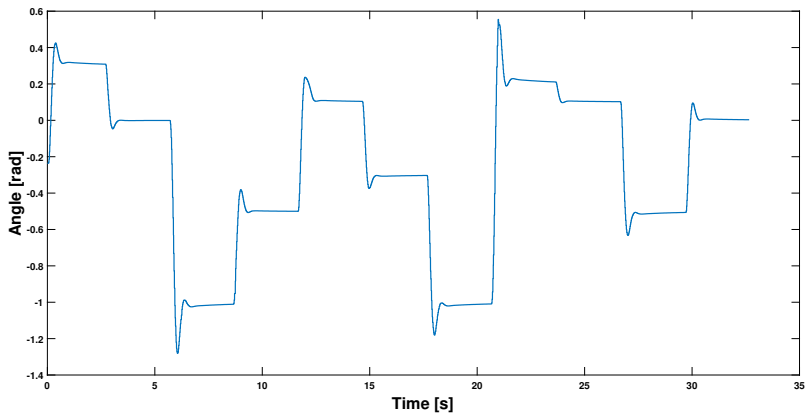


Figure 9.3 Real closed loop step response with different step magnitudes and from different positions (tilt)

9.2 Closed loop sinusoidal response

The results of the sinusoidal response experiments can be seen in Figures 9.4 - 9.6. As the requirement were to track reference signal frequencies up to 1Hz, the 1Hz-case was tested first.

The 1Hz-tracking for the pan-system shows a slight gain in the actual value and of course the time delay which also were seen in the simulation is present, see Figure 9.4. However, the result does correspond to the simulation good enough. Hence, the pan-system fulfills the tracking requirement.

The tilt-system though, does not show a behavior that fulfills the requirement. Why the phenomenon seen in Figure 9.5 happens is not totally clear but a reasonable guess would be that a nonlinearity, possibly caused by gravity or the nonlinear property of the stepper motor, makes it hard for the PID-controller to perform the tracking. In Figure 9.6 the same experiment is shown, but with a reference signal of 0.5Hz. In this case the tracking works to satisfactory.

Due to the problem with the tilt-system, 0.5Hz was considered good enough for this thesis.

The result shows that the assumption done under the system identification phase was not totally correct, i.e. the assumption that both the pan and tilt system had identical dynamics. Unfortunately another model was never investigated due to lack of time.

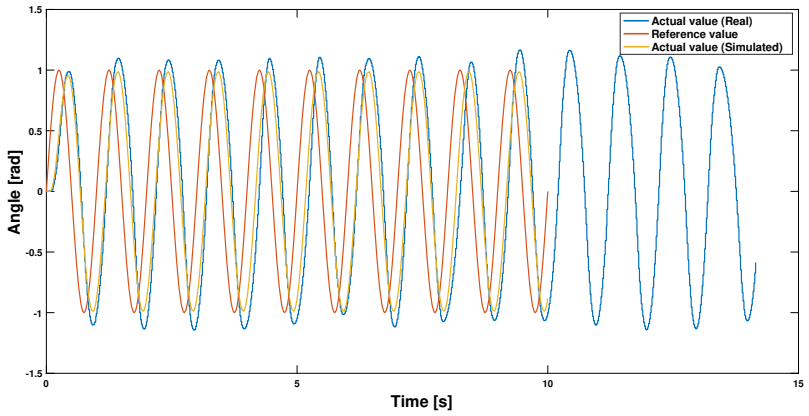


Figure 9.4 Real and simulated closed loop response with a 1Hz sinusoidal reference signal (pan)

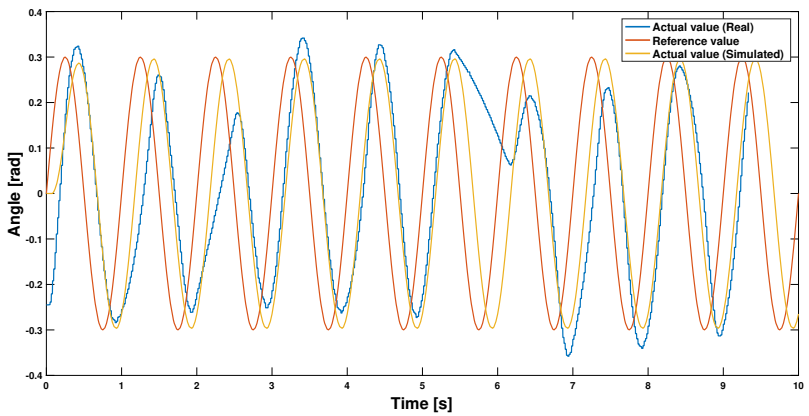


Figure 9.5 Real and simulated closed loop response with a 1Hz sinusoidal reference signal (tilt)

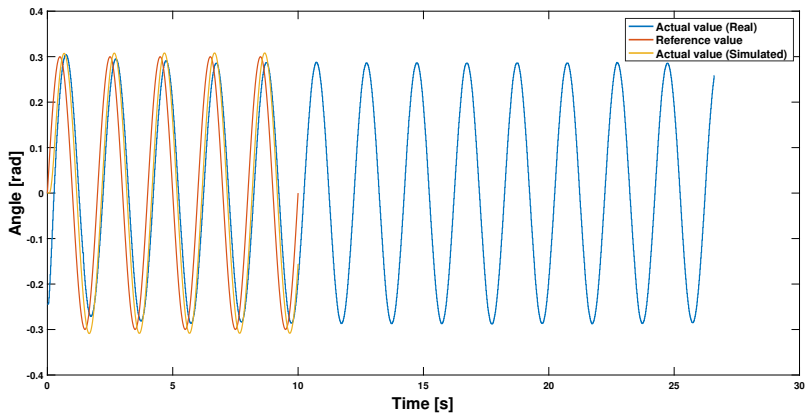


Figure 9.6 Real and simulated closed loop response with a 0.5Hz sinusoidal reference signal (tilt)

9.3 Shake experiments

The shake experiments were carried out in two ways and included the whole system and not just the control part as seen in the two previous sections. The first test was done by mounting the camera on a Shakespear, which is a shaking device that Axis provided, it can be seen in Figure 9.7. The other experiment was carried out in the Robotic lab at LTH. Here the camera was mounted on a robotic arm as seen in Figure 9.8. This experiment made it possible to vibrate the camera with larger amplitudes compared to the Shakespear.

The results of the Shakespear experiment can be seen in Figures 9.9 and 9.10. In this case the Shakespear was set to vibrate sinusoidally in both yaw, pitch and roll with the frequencies 0.40Hz, 0.20Hz and 0.30Hz respectively. The result shows the tracking of the reference signal that is generated to suppress the disturbances. As can be seen, the tracking is not perfect meaning that the disturbances are not completely suppressed. Moreover it is important to keep in mind that the camera lens vibrates a little bit as long as the third reference signal (ϕ_{cRef}), is changing over time. A static value separated from zero will not result in vibrations, but cause an error in desired orientation.

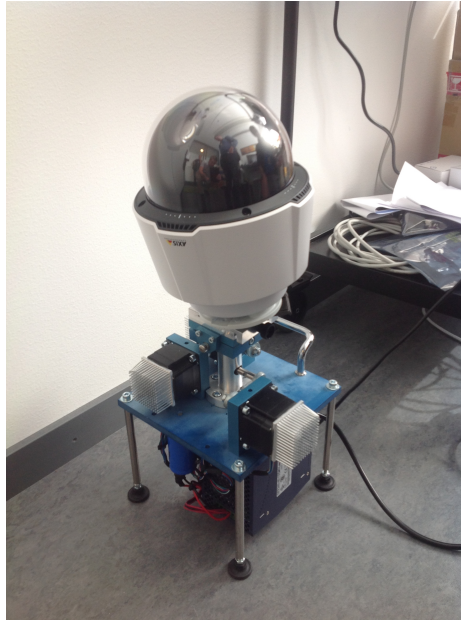


Figure 9.7 The camera mounted on Axis Shakespear

The result of the robotic arm experiment can be seen in Figures 9.11 and 9.12. As can be seen, the amplitude of the disturbances are significantly increased compared to the Shakespear. However, the disturbances change more slowly. Even in this case, the tracking is not perfect. This can be seen more clearly in Figures 9.13 and 9.14 which show the tilt tracking in a smaller scale. As in the simulations the actual value lags a bit behind the reference value. As discussed in the simulation the controller was not optimally designed for ramp references meaning that there can be a better tracking than shown in the figures. Furthermore it can be seen that if a ramp reference makes a sudden change to a constant value there is an overshoot in the tracking.

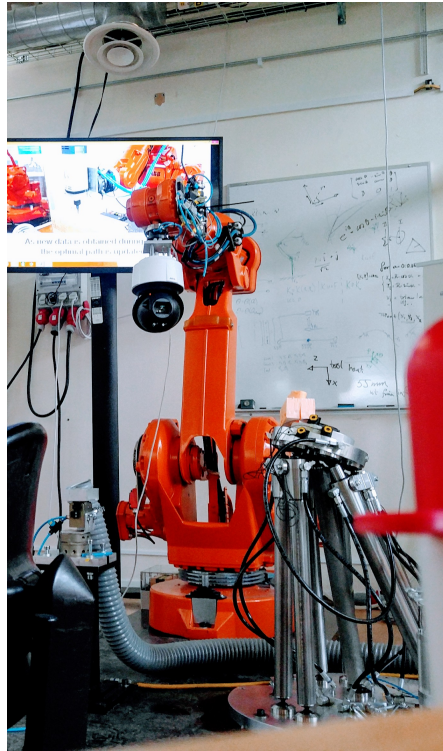


Figure 9.8 The camera mounted on the robotic arm

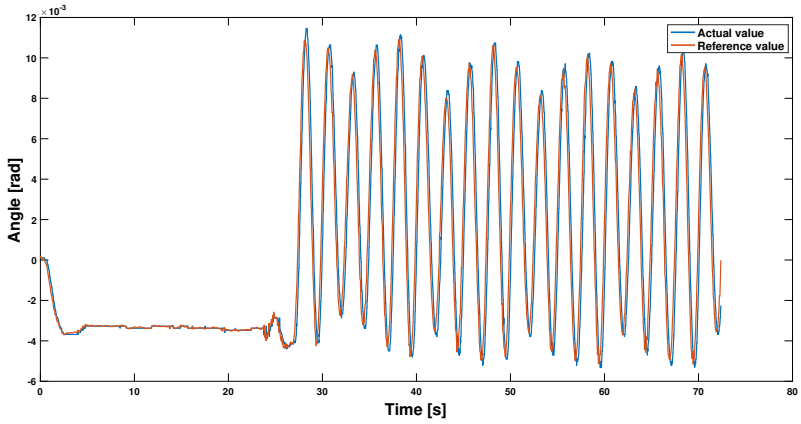


Figure 9.9 Reference tracking (pan) when mounted on Shakespear, with disturbances in yaw = 0.20Hz, pitch = 0.40Hz, roll = 0.30Hz

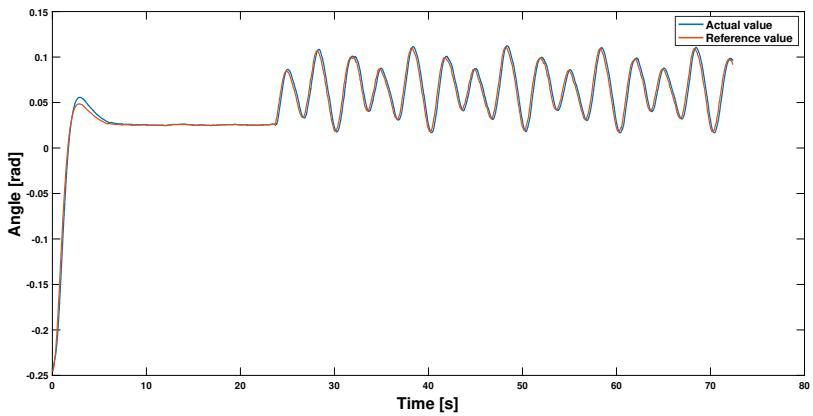


Figure 9.10 Reference tracking (tilt) when mounted on Shakespear, with disturbances in yaw = 0.20Hz, pitch = 0.40Hz, roll = 0.30Hz

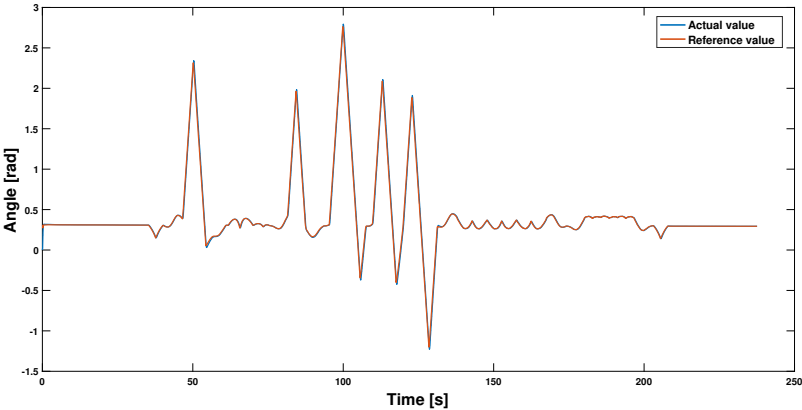


Figure 9.11 Reference tracking (pan) when mounted on moving robotic arm

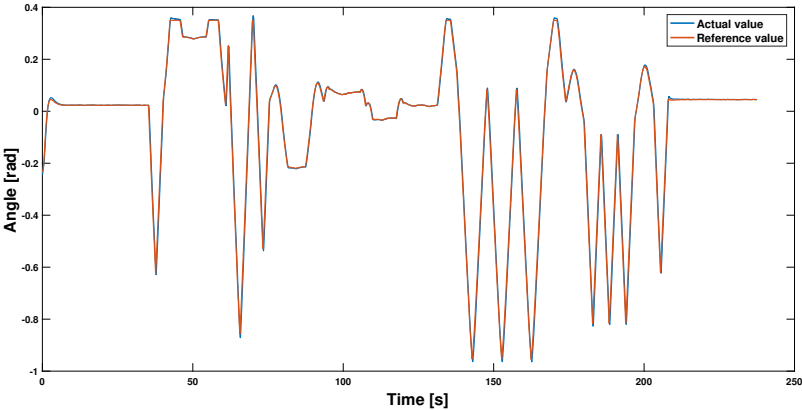


Figure 9.12 Reference tracking (tilt) when mounted on moving robotic arm

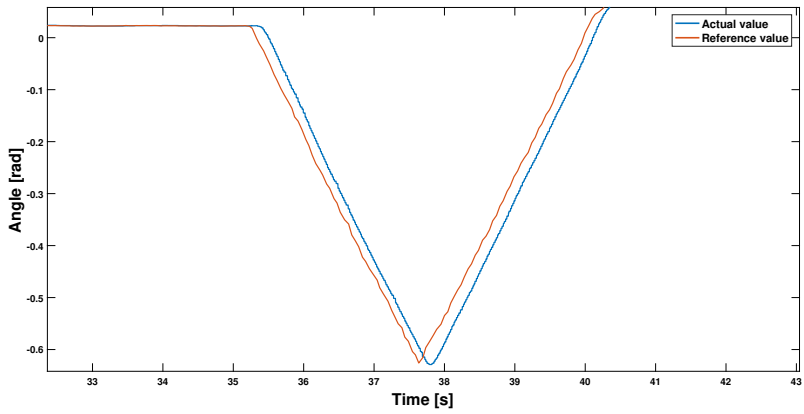


Figure 9.13 Reference tracking (tilt) when mounted on moving robotic arm (Figure 9.12 enlarged)

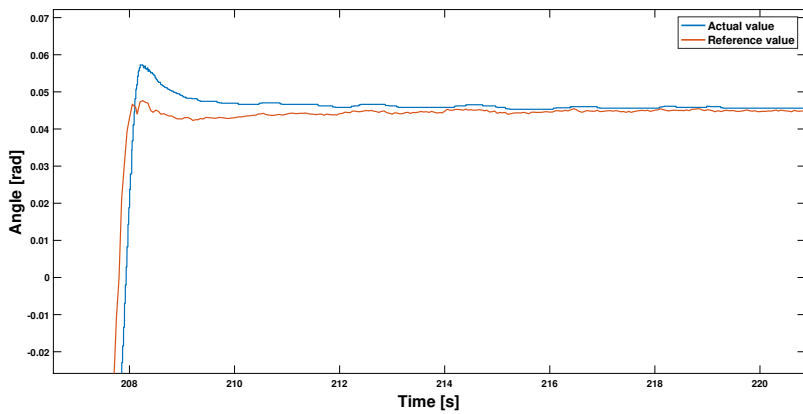


Figure 9.14 Reference tracking (tilt) when mounted on moving robotic arm (Figure 9.12 enlarged)

10

Discussion and conclusions

In this chapter the result of the thesis as whole will be discussed and concluded. Moreover, possible future work will be presented.

10.1 Orientation and sensor fusion

The orientation estimation showed that the Kalman filter gave the best result but on the other hand it is more complex and does thus require more computational capacity. Moreover a great advantage of the Kalman filter is that it can estimate all states in a dynamical model as long as the model is observable.

A problem with the orientation estimation was that the yaw angle relative the NED-frame never could be measured due to the fact that only the gyroscope could be used for that measurement. Another issue was the gyroscope range setting which was set in the smallest possible, i.e. $\pm 250^\circ/s$. This setting can make the gyroscope saturate relatively easy, and this does in turn affect the estimation, and in particular the yaw angle since it cannot be corrected by the accelerometer. However, this was mostly a problem under the test phase where more extreme cases were studied. In a real situation with normal conditions as seen in Figure 3.5 disturbances change more slowly and thus the risk for saturation decreases.

The fact that Euler angles were used for describing orientation introduced a problem called gimbal lock. A way around this problem is to instead use quaternions, i.e. four-dimensional complex numbers, for describing orientation. This was investigated briefly but never tested.

10.2 Reference generation

The reference generator worked as intended in every test that was carried out. Moreover the same generator can be used for cameras that can make roll rotations relative their chassi frame, since a reference signal for that is produced. This makes the designed generator flexible and modular.

10.3 Modelling

The model obtained from system identification was as mentioned in the final results not good enough for the tilt-system. The best approach for the modelling would probably have been to use the motion profile as described in Chapter 7, but this was realized in the end of the thesis.

The physical model is more general and would have been used if no internal motion controller had been present. A particularly strong property of the gimbal model is that it describes the cross-coupling between the systems, and this is not taken into consideration when using the motion profile.

The hybrid stepper motors are as seen in Chapter 6 described by four non-linear differential equations. This non-linearity could have been avoided if, for example, brushless DC-motors had been used instead.

10.4 Control and implementation

The control concept has been shown to work, but the controllers can be better tuned or even changed for another controller type. Gain scheduling can, for example, be tested in order to handle the tilt nonlinearity. In the beginning of the project Model predictive control (MPC) was also considered but was never investigated in depth due to the model uncertainty.

The mid-ranging solution can also be used for future projects since it might be desired to let the electronic image stabilization work together with the mechanical image stabilization.

The code implementation was placed relatively high in the camera code structure. i.e., on the MCPU. If the PID-algorithm had been implemented on the MCU instead delays could have been decreased. This because orientation estimation and reference generation could have been done parallel to the PID-algorithm.

The code that is implemented can also be optimized in order to improve performance.

10.5 Future work

In order to improve the product, the code needs to be refined and a separation of the different systems, i.e. making them work independent of each other, would improve the controllers in particular.

Trying to get the EIS-system, that Axis already have implemented, to collaborate with the mechanical image stabilization will improve the stabilization as it then will be able to handle disturbances with higher frequency. A mid-ranging solution, as the one described in Chapter 8, could perhaps be used for this purpose.

Another aspect that is more control related is to investigate other controller types. As mentioned earlier this can for example be MPC, IMC or gain-scheduled controllers.

Lastly, an investigation of the physical model will most certainly improve the product since the current model is uncertain. Another advantage with this is that the control system then can be implemented in other cameras, with similar construction, more easily.

10.6 Conclusions

To conclude it can be said that almost all the goals that were stated in the introduction were fulfilled. The only goal that was not fulfilled was the requirement of a 1Hz closed-loop bandwidth for the tilt-system. However, the result is good enough to show that mechanical image stabilization is feasible with the Axis camera Q6155-E. The camera was, however, never mounted on a boat but it can be concluded that the stabilization will work as long as the disturbances is within the closed-loop bandwidths. Moreover it is not guaranteed that the stabilization will work for other cameras, since the dynamical model that has been used only holds for the current motion profile configuration. As always, improvements can be made.

Bibliography

- Abdo, M., A. R. Vali, A. Toloei, and M. R. Arvan (2013). “Research on the cross-coupling of a two axes gimbal system with dynamic unbalance”. *International Journal of Advanced Robotic Systems* **10**:10, p. 357. DOI: 10.5772/56963. URL: <http://dx.doi.org/10.5772/56963>.
- Acamely, P. (2002). *Stepping Motors: A Guide to Theory and Practice (Control Engineering) (Control, Robotics and Sensors)*. Institution of Engineering and Technology.
- Alexandre Karpenko David Jacobs, J. B. (2011). *Digital Video Stabilization and Rolling Shutter Correction using Gyroscopes*. Technical Report. Stanford University Computer Science. URL: https://graphics.stanford.edu/papers/stabilization/karpenko_gyro.pdf.
- Alves Neto, A., D. G. A. Macharet, V. A.-c.C.d. S. Campos, and M. F. M. Campos (2009). “Adaptive complementary filtering algorithm for mobile robot localization”. en. *Journal of the Brazilian Computer Society* **15**, pp. 19 –31. ISSN: 0104-6500.
- Axis (2017). Visited: 2017-03-02. URL: <https://www.axis.com>.
- Bilting, U. and J. Skansholm (2011). *Vägen till C*. Lund, Studentlitteratur, 2011 (Ungern). ISBN: 9789144076065.
- CHRobotics (2012). *Understanding Euler Angles*. Tech. rep. Visited: 2017-03-10. URL: <http://www.chrobotics.com/docs/AN-1005-UnderstandingEulerAngles.pdf>.
- Day, M. (2012). “Extracting euler angles from a rotation matrix”. Visited on 2017-05-04. URL: <https://d3cw3dd2w32x2b.cloudfront.net/wp-content/uploads/2012/07/euler-angles1.pdf>.
- Elmenreich, W. (2002). *An Introduction to Sensor Fusion*. Tech. rep. Research Report 47/2001. Institut für Technische Informatik Vienna University of Technology, Austria. URL: https://www.researchgate.net/publication/267771481_An_Introduction_to_Sensor_Fusion.

- Forsman, K. (2005). *Reglerteknik för processindustrin*. Lund, Studentlitteratur, 2005. ISBN: 9144037899.
- Glad, T. and L. Ljung (2003). *Reglerteori : flervariabla och olinjära metoder*. Lund, Studentlitteratur, 2003. ISBN: 9144030037.
- Golik, B. (2006). *Development of a Test Method for Image Stabilizing Systems*. Matr.-Nr. 11038740, Visited: 2017-03-03. Department of Imaging Sciences and Media Technology Cologne University of Applied Sciences. URL: https://www.image-engineering.de/content/library/diploma_thesis/borys_golik_image_stabilizers.pdf.
- Haschke, R., E. Weitnauer, and H. Ritter (2008). “On-line planning of time-optimal, jerk-limited trajectories”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3248–3253. DOI: 10.1109/IR0S.2008.4650924.
- Higgins, W. T. (1975). “A comparison of complementary and kalman filtering”. *IEEE Transactions on Aerospace and Electronic Systems* **AES-11**:3, pp. 321–325. ISSN: 0018-9251. DOI: 10.1109/TAES.1975.308081.
- Kępiński, R., J. Awrejcewicz, and D. Lewandowski (2015). “Dynamical simulation of a nonlinear stepper motor system”. *International Journal of Dynamics and Control* **3**:1, pp. 31–35. ISSN: 2195-2698. DOI: 10.1007/s40435-014-0093-6. URL: <https://doi.org/10.1007/s40435-014-0093-6>.
- Kröger, T. and F. M. Wahl (2010). “Online trajectory generation: basic concepts for instantaneous reactions to unforeseen events”. *IEEE Transactions on Robotics* **26**:1, pp. 94–111. ISSN: 1552-3098. DOI: 10.1109/TR0.2009.2035744.
- Kröger, T., A. Tomiczek, and F. M. Wahl (2006). “Towards on-line trajectory computation”. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 736–741. DOI: 10.1109/IR0S.2006.282622.
- Lambrechts, P., M. Boerlage, and M. Steinbuch (2005). “Trajectory planning and feedforward design for electromechanical motion systems.” *Control Engineering Practice* **13**, pp. 145 –157. ISSN: 0967-0661.
- Lewin, C. (2007). *Mathematics of Motion Control Profiles*. Technical Report. 2017-05-08. Performance Motion Devices. URL: https://performance-motion-devices.s3.amazonaws.com/pdfs/White%20Papers/Mathematics_of_Motion_Control_Profiles.pdf.
- Linderöth, M. (2013). *On Robotic Work-Space Sensing and Control*. PhD thesis TFRT-1098. Department of Automatic Control, Lund University, Sweden.
- Ljung, L. (2007). *System Identification*. Tech. rep. 2809. Linköping University, Automatic Control, p. 65.
- Ljung, L. and T. Glad (2004). *Modellbygge och simulering*. Lund, Studentlitteratur, 2004. ISBN: 9144024436.

- Pedeley, M. (2013). *Tilt sensing using a three-axis accelerometer*. Tech. rep. Visited: 2017-03-27. URL: https://cache.freescale.com/files/sensors/doc/app_note/AN3461.pdf.
- Pycke, T. (2006). *Kalman filtering of imu data*. Visited: 2017-04-14. URL: <http://tom.pycke.be/mav/71/kalman-filtering-of-imu-data>.
- Reflexxes (2017). Visited: 2017-03-06. URL: <http://www.reflexxes.ws/reflexxes-motion-libraries-details.html>.
- ROHM-Semiconductor (2017). *Optical Image Stabilization (OIS)*. Technical Report. Visited: 2017-03-15. ROHM Semiconductor. URL: <http://www.rohm.com/documents/11308/12928/OIS-white-paper.pdf>.
- Sloth Lauszus, K. (2012). “A practical approach to Kalman filter and how to implement it”. *TKJ Electronics*. Visited: 2017-04-26. URL: <http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>.
- Thomas, B. (2008). *Modern reglerteknik*. Stockholm, Liber, 2008 (Slovenien). ISBN: 9147093234.
- Trinamic (2017). Visited: 2017-04-08. URL: <https://www.trinamic.com/>.
- Wittenmark, B., K.-E. Årzén, and K. J. Åström (2014). *Computer Control: An Overview*. Technical Report. Department of Automatic Control, Lund University, Sweden. URL: https://www.ifac-control.org/publications/list-of-professional-briefs/pb_wittenmark_etal_final.pdf/view?searchterm=computer+control%3A+An+overview.
- Zhi, R. (2016). *A Drift Eliminated & Position Estimation Algorithm In 3D*. Master’s thesis 450. University of Vermont, USA. URL: <http://scholarworks.uvm.edu/graddis/450/>.
- Årzén, K.-E. (2014). *Real-time control systems*. Lund University, Department of Automatic Control.
- Åström, K. J. and R. M. Murray (2008). *Feedback systems : an introduction for scientists and engineers*. Princeton, N.J. : Princeton University Press, cop. 2008. ISBN: 9780691135762.

A

Velocity trajectory generation

This appendix contains a more detailed calculation of the velocity trajectory generator, or more specifically the so-called S-profile from Chapter 7. In velocity mode the goal is to reach a target velocity (v_t). When that velocity is reached both the jerk and the acceleration are zero. The procedure is described in Figure A.1 (1A). As can be seen, the target velocity is reached with the help of 3 phases i.e. I, II and III. The goal of the calculation is to obtain the unknown parameters Δt_i which represent phase durations. The limits regarding acceleration and jerk-profile are known. The same holds for the initial values, i.e., a_0 , v_0 , x_0 , and v_t .

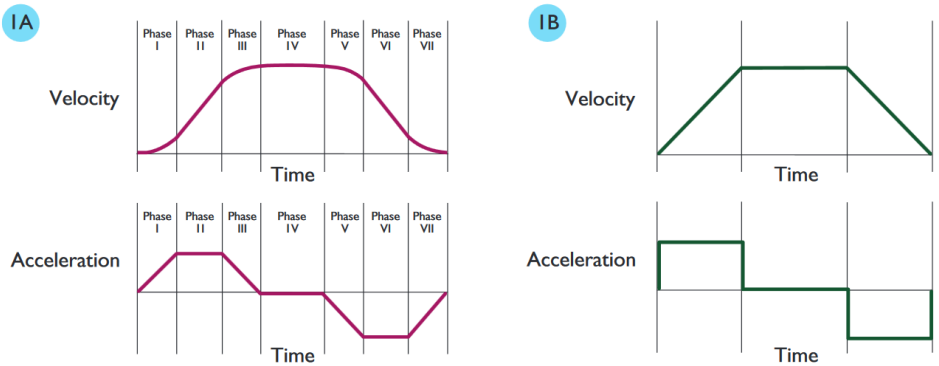


Figure A.1 (1A) S-curve and (1B) trapezoidal profile [Lewin, 2007]

A.1 Velocity mode

This section is based on the ideas presented in Chapter 7, [Kröger et al., 2006], [Refflexes, 2017], [Kröger and Wahl, 2010] and [Haschke et al., 2008], but calculations and notations from [Haschke et al., 2008] are derived and modified to fit the presentation of this thesis.

Equation A.1 sets the direction of the jerk profile, i.e., Equation A.7. v_{const} represents the constant velocity that is reached in phase I, depending on the initial values a_0 and v_0 .

$$d_v = \text{sign}(v_t - v_{const}) \quad d_v \in \{-1, 1\} \quad (\text{A.1})$$

To calculate v_{const} it is necessary to know whether an acceleration or a deceleration is needed for reaching zero in acceleration. This can be found by looking at the sign of a_0 :

$$d = \text{sign}(a_0) \quad d \in \{-1, 1\} \quad (\text{A.2})$$

v_{const} can then be calculated as follows:

$$v_{const} = \frac{a_0^2}{2dJ_1} + v_0 \quad (\text{A.3})$$

There are some cases where the initial value of a_0 is larger than the limit of $d_v a_{max_{acc}}$. In these cases the acceleration needs to be pulled down to $d_v a_{max_{acc}}$ in phase I. This can be done by setting $\pm J_1$ depending on a_0 , $a_{max_{acc}}$ and d_v . Equation A.5 describes the conditions for this:

$$\begin{aligned} d_a &= -1 \quad \text{If: } (a_0 > a_{max_{acc}} \text{ and } d_v = 1) \text{ or } (a_0 < -a_{max_{acc}} \text{ and } d_v = -1) \\ d_a &= 1 \quad \text{Else} \end{aligned} \quad (\text{A.5})$$

Note that $a_{max_{acc}} > 0$.

The duration times of the 3 phases and the corresponding jerk-profile are denoted as

$$\Delta t_i \quad i = \{1, 2, 3\} \quad (\text{A.6})$$

$$\vec{j} = [d_a J_1, 0, -J_3] d_v \quad J_1 > 0 \quad J_3 > 0 \quad (\text{A.7})$$

Constraints & limitations:

$$a_2 = d_v a_{max_{acc}} \quad a_3 = 0 \quad (\text{A.8})$$

Phase profiles:

$$a_i = a_{i-1} + j_i \Delta t_i \quad i = \{1, 2, 3\} \quad (\text{A.9})$$

$$v_i = v_{i-1} + a_{i-1} \Delta t_i + \frac{1}{2} j_i \Delta t_i^2 \quad i = \{1, 2, 3\} \quad (\text{A.10})$$

$$x_i = x_{i-1} + v_{i-1} \Delta t_i + \frac{1}{2} a_{i-1} \Delta t_i^2 + \frac{1}{6} j_i \Delta t_i^3 \quad i = \{1, 2, 3\} \quad (\text{A.11})$$

A.2 Cases

There are two different cases that can appear in a velocity mode calculation. The first case describes how the calculation is carried out when the acceleration is within its limit. The other case describes the opposite, i.e., the calculations when there is a saturation of acceleration.

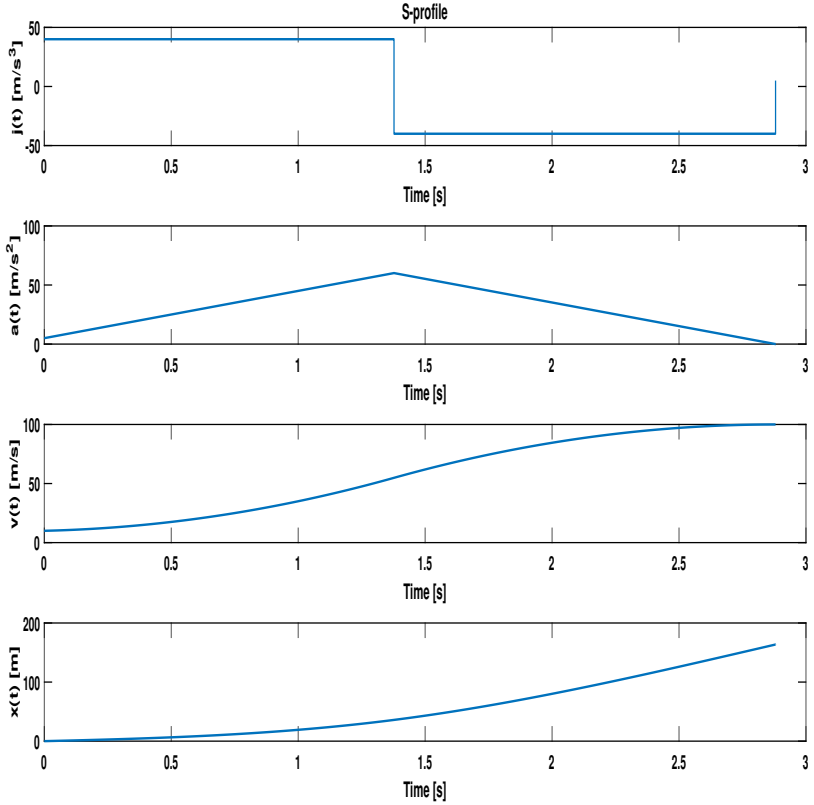


Figure A.2 S-profile in velocity mode when there is no saturation of acceleration

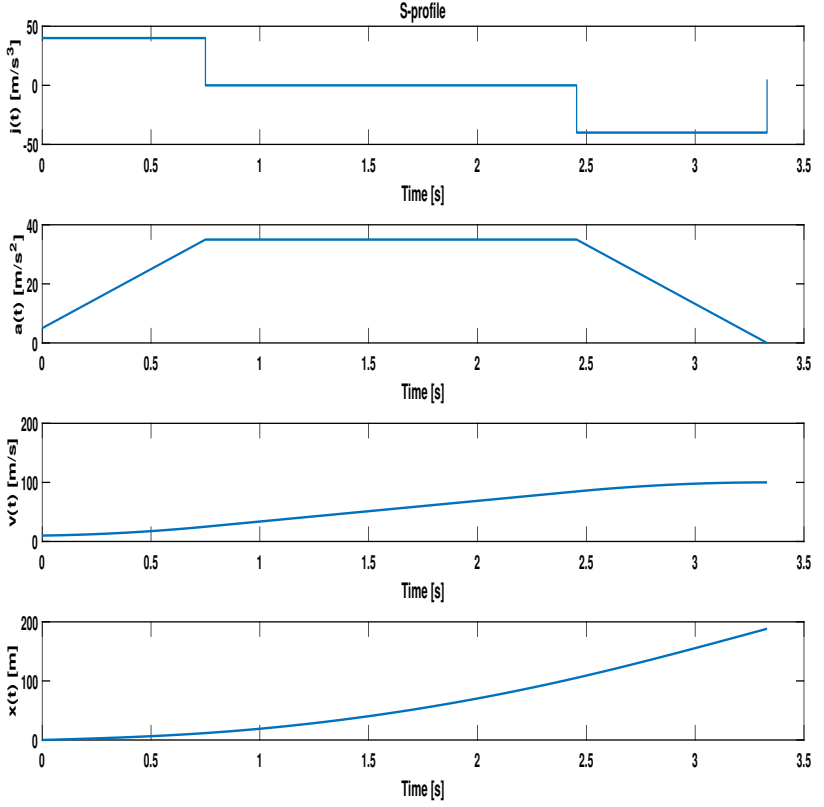


Figure A.3 S-profile in velocity mode when there is saturation of acceleration

Case 1: No saturation of acceleration

When there is no saturation of acceleration, phase II will not exist, because in phase II the acceleration limit is reached. This means that phase I reaches a target acceleration (a_t) but not the maximum acceleration. Phase III has the constraint acceleration equal to zero, and thereby phase III needs to decelerate towards zero. In Figure A.2 this behavior is illustrated. Mathematically, this results in the following:

$$\Delta t_1 = \frac{a_t - a_0}{d_a d_v J_1} \quad \Delta t_2 = 0 \quad \Delta t_3 = \frac{a_t}{d_v J_3} \quad (\text{A.12})$$

By inserting A.12 into A.10 for phase III the following is obtained:

$$v_t = v_3 = \frac{(d_a J_1 + J_3) a_t^2 - J_3 a_0^2 + 2 d_a d_v J_1 J_3 v_0}{2 d_a d_v J_1 J_3} \quad (\text{A.13})$$

If A.13 is solved for a_t two solutions are found. However, there is only one solution for this case, and it depends on the sign of d_v :

$$a_t = d_v \sqrt{\frac{(v_t - v_0) 2 d_a d_v J_1 J_3 + J_3 a_0^2}{d_a J_1 + J_3}} \quad (\text{A.14})$$

The solution from A.14 needs to be checked with A.15. If A.15 is true the assumption regarding a_t is wrong and Case 2 needs to be solved.

$$|a_t| > a_{\max_{acc}} \quad (\text{A.15})$$

Note that Case 1 is not valid if the denominator of Equation A.14 is zero i.e. $d_a J_1 + J_3 = 0$. If this is the case, Case 2 needs to be solved.

Case 2: Saturation of acceleration

In this case the acceleration is saturated, as seen in Figure A.3, and thereby the acceleration target a_t is known, i.e., it is equal to the maximum acceleration. Thus, both Δt_1 and Δt_2 are known from the following:

$$\Delta t_1 = \frac{a_t - a_0}{d_a d_v J_1} \quad \Delta t_3 = \frac{a_t}{d_v J_3} \quad a_t = d_v a_{\max_{acc}} \quad (\text{A.16})$$

In order to get Δt_2 , Equation A.10 can be used for phase III. This gives the following:

$$v_t = v_3 = v_0 + a_0 \Delta t_1 + \frac{1}{2} d_a d_v J_1 \Delta t_1^2 + (a_0 + d_a d_v J_1 \Delta t_1) \Delta t_2 - \frac{1}{2} d_v J_3 \Delta t_3^2 + \Delta t_3 (a_0 + d_a d_v J_1 \Delta t_1) \quad (\text{A.17})$$

If A.17 is solved for Δt_2 the following is obtained:

$$\Delta t_2 = \frac{v_t + \frac{1}{2} d_v J_3 \Delta t_3^2 - \frac{1}{2} d_a d_v J_1 \Delta t_1^2 - a_0 \Delta t_1 - v_0}{a_0 + d_a d_v J_1 \Delta t_1} - \Delta t_3 \quad (\text{A.18})$$

Lund University Department of Automatic Control Box 118 SE-221 00 Lund Sweden		<i>Document name</i> MASTER'S THESIS	
		<i>Date of issue</i> September 2017	
		<i>Document Number</i> TFRT-6042	
<i>Author(s)</i> Magnus Gustavi Louis Andersson		<i>Supervisor</i> Björn Ardö, Axis Communications Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden Karl-Erik Årzén, Dept. of Automatic Control, Lund University, Sweden (examiner)	
<i>Title and subtitle</i> Implementation of control algorithm for mechanical image stabilization			
<i>Abstract</i> <p>Cameras mounted on boats and in other similar environments can be hard to use if waves and wind cause unwanted motions of the camera which disturbs the desired image. However, this is a problem that can be fixed by applying mechanical image stabilization which is the goal of this thesis.</p> <p>The mechanical image stabilization is achieved by controlling two stepper motors in a pan-tilt-zoom (PTZ) camera provided by Axis Communications. Pan and tilt indicates that the camera can be rotated around two axes that are perpendicular to one another.</p> <p>The thesis begins with the problem of orientation estimation, i.e. finding out how the camera is oriented with respect to e.g., a fixed coordinate system. Sensor fusion is used for fusing accelerometer and gyroscope data to get a better estimate. Both the Kalman and Complementary filters are investigated and compared for this purpose. However, the Kalman filter is the one that is used in the final implementation, due to its better performance.</p> <p>In order to hold a desired camera orientation a compensation generator is used, in this thesis called reference generator. The name comes from the fact that it provides reference signals for the pan and tilt motors in order to compensate for external disturbances. The generator gets information from both pan and tilt encoders and the Kalman filter. The encoders provide camera position relative to the camera's own chassi. If the compensation signals, also seen as reference values to the inner pan-tilt control, are tracked by the pan and tilt motors, disturbances are suppressed.</p> <p>In the control design a model obtained from system identification is used. The design and control simulations were carried out in the MATLAB extensions Control System Designer and Simulink. The choice of controller fell on the PID.</p> <p>The final part of the thesis describes the result from experiments that were carried out with the real process, i.e. the camera mounted in different setups, including a robotic arm simulating sea conditions. The result shows that the pan motor manages to track reference signals up to the required frequency of 1Hz. However, the tilt motor only manages to track 0.5Hz and is thereby below the required frequency. The result, however, proves that the concept of the thesis is possible.</p>			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>	
<i>Language</i> English	<i>Number of pages</i> 1-94		
<i>Security classification</i>			