DEGREE PROJECT, IN PROGRAM , FIRST LEVEL
STOCKHOLM, SWEDEN 2015

The Black Box

CONTROL AND DESIGN OF REACTION WHEEL "BASED"ADASD INVERTED PENDULUM

ALEXANDER RAMM & MIKAEL SJÖSTEDT

KTH ROYAL INSTITUTE OF TECHNOLOGY

KTH SKOLA

The Black Box

Control and design of reaction wheel based inverted pendulum

MIKAEL SJÖSTEDT
ALEXANDER RAMM

Bachelor's Thesis in Mechatronics

Supervisor: Daniel Frede
Examiner: Martin Edin Grimheden
Approved: TBA 2015-month-day

TRITA xxx yyyy-nn

Abstract

Some robots can do remarkable things. Balancing something unstable is one of those. This paper covers the design of such a robot and evaluates how sensor placement affects the ability to balance.

From the result it was clear that some placements were bad and clearly affected the performance of the robot. The best placement was the farthest one from the motor, probable thus less affected by the induce currents in the motor windings.

OLD abstract Robots that move require a high degree of precision from position tracking sensors. This paper studies how the placement of these sensors effect the robots ability to determine its position. A robot with a cubical frame is built, which were able to balance on a edge with help of a reaction wheel. The robot determined its rotation using an inertial measurement unit. Different sensor positions were evaluated empirically and...

More selling from beginning. Dont miss results and conclusions

Sammanfattning

Stabilisering med svänghjul Utevkcla...

Robotar som förflyttar sig kraver mycket precis nogrannhet från sina positionerings sensorer. Det här rapporten tar upp hur placeringen av dessa sensorer påverkar robotens förmåga att bestämma sin position. Från en kubformad ram byggdes en robot, som med hjälp av ett motordrivet svänghjul kan applicera ett internt moment för att balansera på en kant. Roboten använde en sensor av typen –inertial measurment unit– för att bestämma sin position. Olika placeringar av sensorn utvärderases empiriskt och ...

Preface

We would like to thank Daniel Frede, our supervisor, for his continous support. Always ready with a well filled beer glass and pinpointing our faults with *German precision.* Staffan Qvarnström and Tomas Östberg for their mechanical and practical expertise. As well all the student assistants that have helped us throughout the course.

Alexander Ramm
Mikael Sjöstedt
KTH, may, 2015

Contents

Nomenclature

Symbols - needs restructure

Symbol	**Description**
E	Elasticity module (Pa)
r	Radius (m)
t	Thickness (m)
$\mathcal{L}$	Lagrange
θ	Cube angle
ϕ	Flywheel angle
q	Lagrange operator
E_k	Kinetic energy
E_p	Potential Energy
I_c	Inertia of the cube
I_f	Inertia of the flywheel
M_{tot}	Total mass of the cube
M_f	Mass of the flywheel
i	Current
K_t	Motor torque constant
E_{emf}	Induced voltage
K_{emf}	Motor voltage constant
U	Voltage across motor poles
R_m	Motor internal resistance
η_m	Motor efficiency
η_g	Gear efficiency
Γ	Gear ratio
z	Measurement noise
w	Process noise

Abbreviations - Alphabetical order

Abbreviation	**Description**
CAD	Computer Aided Design
CAE	Computer Aided Engineering
PLM	Product Lifecycle Management
PWM	Pulse With Modulation
DOF	Degrees of freedom
MEMS	Microelectromechanical Systems
MATLAB	Matrix Laboratory, computational program
RMS	Root Mean Square
MCU	Microcontroller
IC	Integrated circuit
I^2C	Inter-Integrated circuit
USB	Universal Serial Bus
UAV	Unmanned Aerial Vehicle
IDE	Integrated Development Interface
DC	Direct Current
dps	Degrees per second

Chapter 1

Introduction

“Begin at the beginning," the King said, very gravely, "and go on till you come to the end: then stop.”

Lewis Carroll, Alice in Wonderland

This chapter describes the background, purpose and scope of this project conducted at the mechatronics department at the Royal Institute of Technology, KTH, Sweden. The work was carried out during the spring 2015.

1.1 Background

A reaction wheel is a wheel that is accelerated to apply torque. The most wide spread use of reaction wheels is in human made satellites. The reaction wheels, usually three of them in the case of satellites, are used to change the attitude of the satellite by applying torque in a favourable manner. This is imperative to direct solar panels towards the sun or pointing antennas to assure maximum performance and connectivity to the satellite. Satellites are quite uncommon (there are just over thousand satellites currently in orbit), and their technology can at times seem alien. But reaction wheels should not be alienated, they can be used in many contexts and this paper will cover one of them.

Balancing 1 *degree of freedom* (DOF) inverted pendulum type structures using reaction wheels is no new concept, and became more accessible with the introduction of cheap microcontrollers. The use of automated control is growing at a rapid pace and is being implemented more and more in consumer related products. This growth has made automated control together with sensors available more now than ever. It can be seen in the every-day life in product lines such as mobile phones, gaming controllers, cars and UAVs such as quadrocopters.

One of the most basic systems that requires some control to become stable is the inverted pendulum. Although it is simple to define, its control is not. A lot of work has been done on the topic but there are still a limited amount of knowledge available to the public.

The method to achieve balance of the pendulum using reaction wheels is even more narrow. In recent years prototypes of land based structures using reaction wheels have been a hot topic and some of these robots are truly remarkable. It would be a great achievement for the authors to contribute knowledge about how such a mechanism could be built and evaluate the capabilities and restrictions of such a machine. **sista stycket ?**

1.2 Purpose

The goal of the project was to build a structure with one degree of freedom that can maintain balance using a reaction wheel and examine the behaviours of the system. A concept idea of the robot can be seen in Figure 1.1

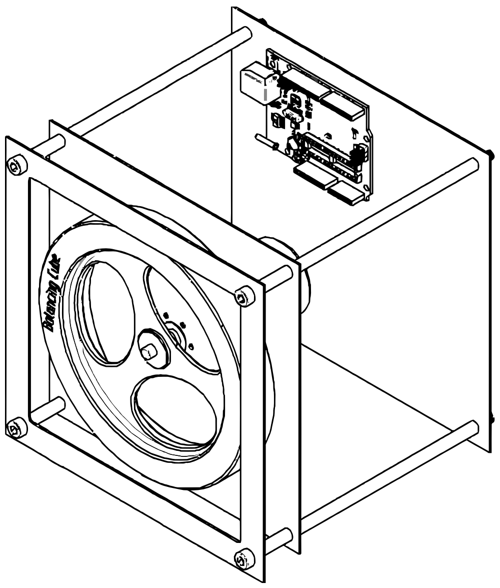


Figure 1.1: Concept design

The dynamics is mostly effected by the control system, which is responsible for accelerating the motor in the correct angular direction, to maintain balance. The parameters in the control system effects response time, overshoot and sinusoidal settling time. This project will hopefully contribute to some development within the open-source community. All results are available online, open source on GitHub [Sjöstedt and Ramm, 2015]. As a thesis in mechatronics, this paper can be divided into two parts. One engineering part whose focus to implement knowledge in mechanics, electronics and control theory to result in a functioning robot. A research part whose topic could be concentrated to a question

How does the sensor placement effect the quality of the sensor data. — what is quality ? how it effects system performance?

With quality defined as the usability of the data given by the sensor. **somewhat vague**
The sensors used for this project is an *Inertial Measurement Unit* (IMU) which can be placed arbitrarily. Certain positions might have an advantage in terms of how usable the raw data is. The IMU is a sensitive devise and disturbances such as high current and fast oscillations in its vicinity might effect the data, which is to be investigated.

1.3 Scope

The only sensor whose position is to be examined is the IMU. Only a few key positions of the sensor is going to be examined.

The effects that were looked at were the ones related to the control system. Mainly the overshoot behaviour and the settling time of the system. Only data used for the specific control system were examined, **other DOF measurements were not taken into consideration ???**. I.e. **its a consquence** the results may only be applicable in similar machines and not in general.

1.4 Method

Initially the sensor was placed in the upper corner as far possible from other components, **ja det ska ju utvecklas...** one of the side corners, between the mentioned corners and in the center of the cube (insert figure reference). Both raw data and filtered data were collected and sent over serial to a computer. Using Matlab [MATLAB, 2014] the readings where analysed for stablization behaviour. Other obvious observations was noted. To have equal conditions, all measurements were made on the same horizontal space with the same external voltage supply.

For every position the same parameters and constants were used in all systems and software. The comparisons were made in between measurements while balance was maintained and no external disturbance is applied. All measurements where taken during a limited time frame (We dont know this yet....). **this might belong to method?**

New TRY The cube was placed at rest close to the balance point. The support was removed and the constrol system enabled. The time taken for the c´ube to fall on its side were measured. If the cube did not fall within five seconds the position were called stable.

Five different positions were examined, they can be seen in figure **In MEd ny bild här**. Ten measurement were taken for every position and the average falltime for the particular position was calculated.

Chapter 2

Theory

This chapter cover some of the theory that is required if one want to build a similar robot. It is assumed that the reader has some understanding of Newtonian mechanics, signal analysis and control theory. Basic understanding of DC motor operation is also an advantage.

The first part is about the inertial reference unit that covers issues of sensor characteristics and why they are important to the system as a whole and the research question in particular.

There is a part that discuss Kalman filter theory, a filter required for retrieving high quality data from the IMU. The filter interprets noisy data from the sensor and digitally filters the signal to a more trustworthy output.

The **last part** covers the theory of the mechanical system behaviour that is used to develop the state space control system. The equations are responsible for the actual balance part and thus important.

2.1 Inertial Measurement Unit

2.1.1 Background

The data collected for calculating the angle of the cube is gathered from an IMU, a unit that uses both an accelerometer and a gyroscope to track the orientation and position. A IMU is often rated for several *degrees of freedom* (DOF), a unit specified as 6-DOF uses three orthogonal accelerometers and gyroscopes. These measure linear acceleration and angular velocity respectively. There are also units that are rated for additional degrees of freedom and usually include features such as magnetometer or barometer sensors. To understand the fundamentals of an inertial system a Cartesian coordinate system is defined in Figure 2.1

The accelerometer is used to measure the acceleration along the lines of x, y, z axes while the gyroscope measures the angular rate around these axes.

The inertial navigation system used in this project is a small *microelectromechanical system* (MEMS). A micromechanical sensor is a very small unit that make use of its mechanical properties to sense alteration in the environment [An-

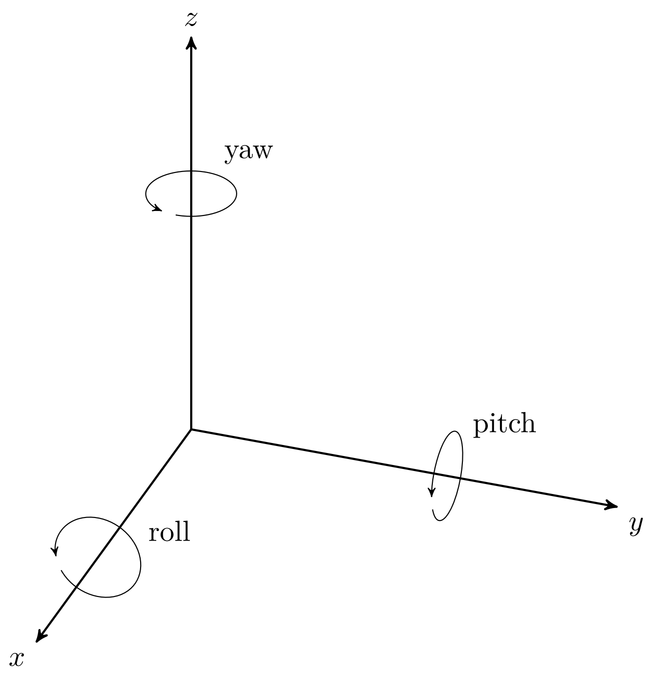


Figure 2.1: Cartesian coordinate system

drejasic, 2008]. The advantages of these small units are low production costs, small size, low power consumption and good accuracy. But all these advantages comes at a price, which is signal noise [Woodman, 2007].

2.1.2 Accelerometer

Accelerometers are used to measure linear acceleration. Or rather, it measure forces due to acceleration. These forces can be divided into two groups

- Static forces, such as gravity
- Dynamic forces, due to movement

The force is then converted into acceleration, this is done by measuring the change in capacitance when a spring mass system is moving. The typical accelerometer consists of a movable mass that is attached via a mechanical spring or suspension system to a frame that is used as reference. The capacitance is usually measured between a conductive plate and the known mass, seen in figure 2.2. The capacitance is then used to determine the displacement of the mass and then calculate the force applied to it with the known spring mass system.

Typical noise sources in accelerometers are mechanical vibration of the springs, circuitry and general equipment disturbances. The accelerometer in this project is used to cross-check the perceived angle by the gyroscope, i.e. the signal has to be

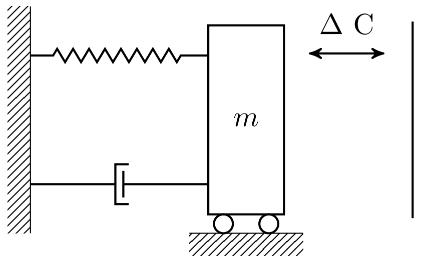


Figure 2.2: Accelerometer capacitance

integrated to compute the angle. To specify how much the noise effects the integrated value is represented by the *velocity random walk* (VRW) [Stockwell, 1997]. The signal from the accelerometer during a stationary position can be characterized as a white noise, and the integrated value is going to "walk" around the mean value, hence the name *random walk*. The accelerometer also outputs a bias, it is essential to determine the bias when estimating a position otherwise the error would grow quadratic due to multiple integration of the measurements [Andrejasic, 2008].

2.1.3 Gyroscope

Gyroscopes unlike accelerometers, do not measure linear acceleration. Gyroscopes measure the angular velocity. This is done by making use of the Coriolis effect to measure the angular rate.

Consider figure 2.3 where a mass is vibrating along the y-axis, with the momentary velocity v. When the mass is rotated along the z-axis with the angular velocity w, a secondary vibration perpendicular to the first is induced which is explained by the Coriolis force

$$\mathbf{F}_c = -2m(\boldsymbol{w} \times \boldsymbol{v}) \tag{2.1}$$

The result is a physical displacement and a capacitance is measured just like the accelerometer.

A micromechanical gyroscope is, like the accelerometer, effected by a bias. This is often due to friction caused by moving parts or production variations induce stress on the construction resulting in an offset of the output. If a constant error is integrated the angular error grows linearly with time. This is easily corrected by subtracting the bias from the output. This approach has a drawback. The small size and sensitivity of this device make the bias wander due to flickering noise in the electronics [Woodman, 2007]. Thus a *bias stability* is introduced as a measurement of how the bias may change during a period of time. Other troublesome errors that occur in MEMS gyroscopes are thermo-mechanical white noise similar to noise in an accelerometer. To indicate how this noise effects the integrated value an *Angle Random Walk* (ARW) is introduced. The ARW corresponds to the VRW mentioned in section 2.1.2 but considers the angular velocity instead.

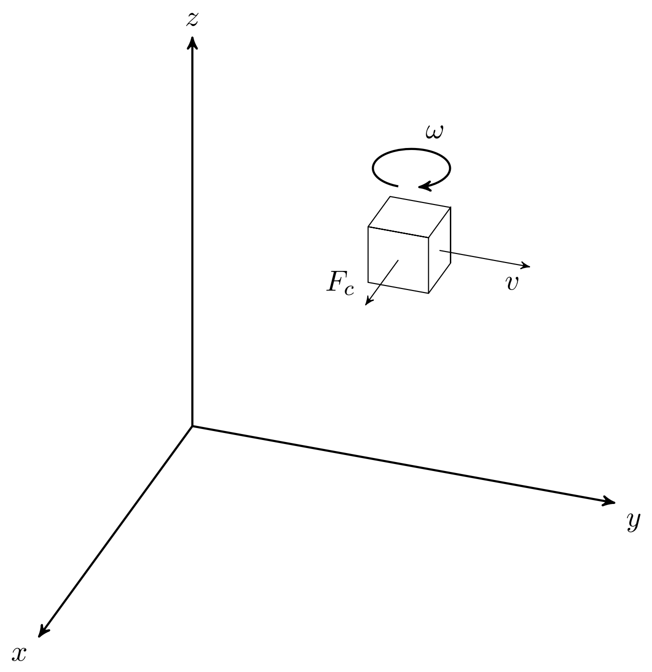


Figure 2.3: Gyroscope pic

The concepts of ARW, VRW and bias stability that has been introduced are more or less an indication of how precise the measurement devices are.

2.2 Kalman filter

2.2.1 Introduction

The signal from an IMU contains data of angular velocities and acceleration, but also a lot of noise, as explained earlier. A position estimated from an untreated signal from an IMU could work for short periods, but over time the estimated position *drifts* [Shiau et al., 2012]. This drift occurs when measurements containing noise is integrated to acquire a position, the readings contain both white noise and often a bias which is making the error to grow for every calulation. Integrating the angular motion from the gyroscope to estimate a position would result in an angular drift and an even worse drift for the accelerometer as it is integrated twice to estimate a position. By using a Kalman filter the drift can effectively be minimized. If the readings from both the gyroscope and accelerometer is considered, and with some help of probablity theory the estimated state is not far from the true value. A Kalman *filter* is not what the name suggests, it is an estimator. Old and new measurements are processed real-time to calculate an estimation of the current state.

One could say that the Kalman filter moves between TWO PHASES (NO GOOD) see Figure 2.4. Firstly it estimates a state using a known process, which usually

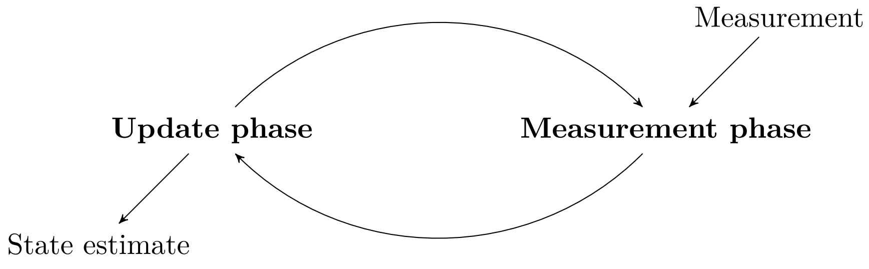


Figure 2.4: Kalman phases

contains a noise. Then it adjusts the state to a measurement.

Keep in mind that there are some regards that should be taken into consideration when choosing an estimator. A good estimator produces states that are non biased, *values that have an average of the true value.* As well that the estimated state variance from the true state is as small as possible[Simon, 2001].

2.2.2 State Estimator

The Kalman filter is, as stated above, a state based estimator. By using the last measurement and the one before that it can derive a better estimate of the current state. The true state and the measured value at a time k would be

$$x_k = Ax_{\text{k-1}} + Bu_{\text{k-1}} + w_{\text{k-1}} \tag{2.2}$$

$$z_k = Hx_k + v_k \tag{2.3}$$

The true state x is expressed with the the old state and an input u. But the signal also contains a process noise w. The process noise w in equation (2.2) is a representation of variances in the process that cannot be mathematically predicted. When using a gyroscope this reflects the error characteristics mention in section 2.1.3. The measured value, z seen in equation (2.3) is an observed measurement. Ideally this would only be a function of x, but is distorted by the measurement noise v. The measurement noise, v, much like the process noise is common in any measurement and represents various fluctuations caused by the equipment.

The Kalman filter takes into consideration how reliable the process and measurement values are using the covariance error matrices described in A.1. These help the filter a reliable state $\hat{x}_k$

$$\hat{x}_k = \hat{x}_k^- + K_k\tilde{y}_k \tag{2.4}$$

where $\hat{x}_k^-$ is an estimation of the state using the process, in this case the gyroscope. The second term $K_k\tilde{y}_k$ contains $\tilde{y}_k$ which is a **parameter** that consider the measurement, the accelerometer. It also contains the Kalman gain, K_k which indicates how reliable the measurement is and how much it should effect the new estimated state

For further reading, and mathematical proof see [Welch and Bishop, 2006].

2.3 Model dynamics

To create a control system, the physical model has to be translated to a mathematical model. The system can be estimated much like a two-degree-of-freedom inverted pendulum model [Chauveau et al., 2005].

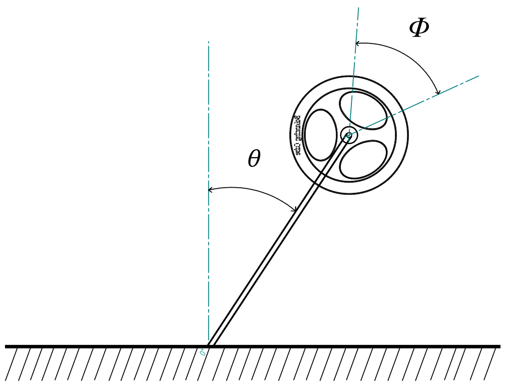


Figure 2.5: Cube modelled as a reaction wheel pendulum

Lagrangian Dynamics have been used to derive the systems behaviour. For a described mathematical walkthrough of how the model has been derived see appendix B.1. First of all cube's movement θ is expressed by the flywheel movement ϕ. The Lagrangian results in a torque **equation** around the pivot point

$$M_{tot} \cdot g \cdot l \cdot \sin\theta - I_c \cdot \ddot{\theta} = I_f \ddot{\phi} \tag{2.5}$$

By making the assumption that the cube will balance with small deviations from the equilibrium point the angle can be expressed without the sinusodial term. By transforming equation 2.5 to the Laplace domain it is possible to express a transfer function for the flywheel angle

$$\phi(s) = \frac{M_{tot} \cdot g \cdot l \cdot -I_c \cdot s^2}{I_f \cdot s^2} \theta(s) \tag{2.6}$$

As the controller is regulated by voltage as input a relation between voltage and flywheel angle has to be determined. The torque exerted on the flywheel depends on the current flowing through the motor as well as gearing and efficiency of the components

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \cdot i \tag{2.7}$$

Using the voltage across the motor poles the equation 2.7 can be written as

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \frac{U - \dot{\phi} K_e}{R} \tag{2.8}$$

By transforming equation 2.8 to the laplace domain tand combining it with equation 2.6 a complete transferfunction can be derived which express how the cube's angle depends on the input voltage in the Laplace domain

$$\theta(s) = \frac{K_t \cdot I_f \cdot \eta_m \cdot \eta_g \cdot \Gamma \cdot s^2}{R_a(M_{tot} \cdot g \cdot l - I_c \cdot s^2)(I_f \cdot s^2 + \frac{K_t \cdot K_e \cdot s}{R_a})} U(s) \tag{2.9}$$

With the transferfunction from equation 2.9 **ungefärliga pid värden?** proper PID-parameters can be derived for the demonstrator

2.4 Control theory and Pole placement

When using a state-space control system the poles of the controller is chosen by the engineer. But if a poor choice is made the response of the system will not satisfy the requirements of the regulator. If place to far away from the openloop poles high input signals might be required [Glad and Ljung, 2006]. The openloop poles can be calculated by solving the eigenvalues for the A matrix. Also one must check that the system is controllable for the posed states. Then a gain matrix can be computed and the system can be closed.

2.5 Pulse Width Modulation

Flytta till demonstrator? Pulse Width Modulation: According to en.wikipedia.org: "Pulse-width modulation (PWM) [...] is a technique used to encode a message into a pulsing signal." This is useful in many power applications, from dimming LED's to motor control. The idea of PWM is to alter the voltage over a device while still only having a set level voltage to supply. Simplified the voltage is cut into a square wave. This is done by using two transistors, one that can short the motor poles and one that distributes the supply voltage to be applied to the motor [Johansson, 2003]. If these transistors are switched on and of fast compared to the time constant of the motor, the root mean square (RMS) voltage will be the acting voltage. This allows for a DC voltage that is perceived as lower than the acual supply voltage. They duty cycle can be changed during operation unlike liner type voltage regulators.

Chapter 3

Design and implementation

This part covers the design of the demonstrator, as well as how the theory introduced in chapter 2 was implemented.

3.1 Problem Formulation

The construction of the cube can be seen as the engineering problem. The cube should be a robot that, using a reaction wheel can balance on its edge. All components were to be mounted in the robot, then only requiring a power source.

The main components was a frame, reaction wheel. motor, motor controller, Arduino, and IMU

The main goal of this project was to build a structure which remain stable in an unstable condition. A process of this sort can be divided into several parts.

- Design
- Motor Control
- Sensor Reading
- System Control

All these individual systems had to be implemented and joined in the final assembly.

3.2 Design

3.2.1 Frame

The main construction problem was to decide the size of the cube and reaction wheel. A heavier construction with more inertia results in a pendulum which is simpler to stabilize. But also demands a larger torque and more powerful motor controller. The of the cube was constructed in aluminium consisting of 3 **symmetric** plates. By keeping the symmetric parts heavy considering the electrical components the subsystems could be placed arbitrarily and the center of gravity assumed to be in the center of the cube. By constructing the cube in such a way the mathematical model was easier to derive. The final design is seen in Figure 3.1. The motor was fixed through the middle wall, the shaft on one side and the body on the other. The motor controller and microcontroller fastened at the backplate whilst the sensor is placed on the front, to be moved during the research.

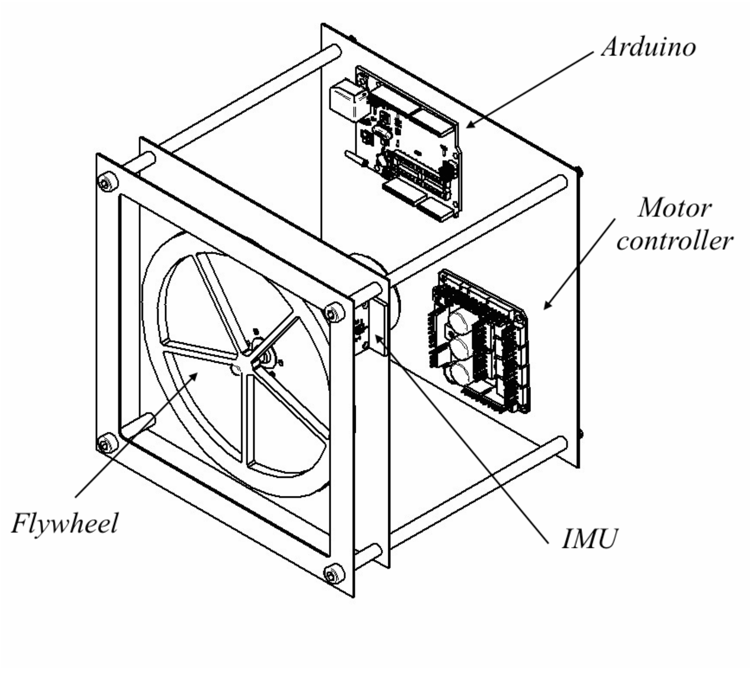


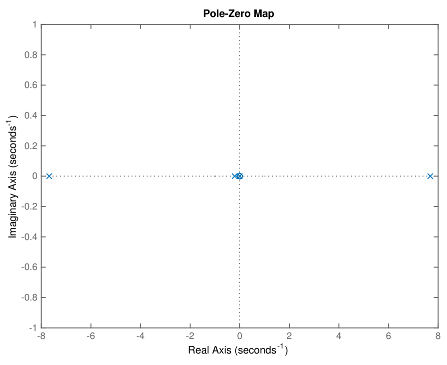
Figure 3.1: The final design of the cube

3.2.2 Arduino

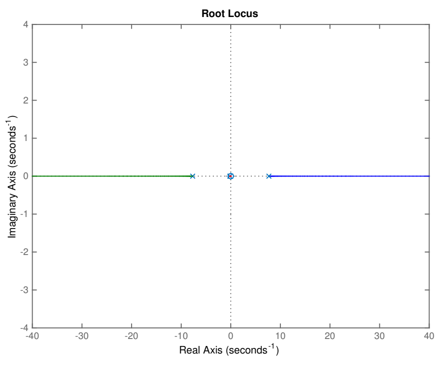
The arduino is a open source, 8-bit microcontroller that was used in the robot. The board run on 5 and 3.3 volts and has outputs for both. This was used to power both the IMU and the motor controller, external power only had to be used for the motor main supply. The arduino has 6 PWM ports, which can be implemented with a singe function call each.

3.2.3 Motor controller

A factory made full H-bridge DC motor controller on a chip were used. The main criteria the controller had to meet was current and voltage capabilities suitable for the motor. The controller had to be able to deliver enough current to the motor



(a) Pole Zero map



(b) Root Locus

Figure 3.2: A figure with two subfigures

when a load were applied. A motor controller from Pololu was used specified at 28V and able to handle continous currents of three ampere. The chosen controller also had a safety function that disabled the motor if an over current ran through the controller.

It was connected to the Arduino, with 4 cables each for a different functionality. PWM signal input, a brake function input, a direction switch input and an status flag output. A schematic diagram of all these connections can be seen in figure x. **in med fixbit bild här**

3.2.4 Motor

To apply torque to the flywheel a Maxon DC motor with a nominal voltage of 24V was used. **direkt kopplat lulul?** The torque excerted on the flywheel is straight connected to the ability to balance the cube at a larger deviation from the center-line. To gain a large torque but still maintaining a current that the motor controller could handle a spur gear with a ratio of 10 was used.

3.3 System analysis

To analyse how the system behaves the transfer function obtained from equation 2.9 is used

$$G(s) = \frac{K_t \cdot I_f \cdot \eta_m \cdot \eta_g \cdot \Gamma \cdot s^2}{R_a(M_{tot} \cdot g \cdot l - I_c \cdot s^2)(I_f \cdot s^2 + \frac{K_t \cdot K_e \cdot s}{R_a})} \tag{3.1}$$

Figure 3.2 shows the Pole-Zero map and Root locus of the system indicating that the system is unstable in the open loop system. [Glad and Ljung, 2006] To evaluate if the closed loop system is stable with a scalar gain K, Root Locus in Figure 3.2b is used which indicates that the system is unstable with a loop gain K.

The input signal U i.e. voltage supplied to the motor is limited therefore a linear quadratic regulation is used. This is done with Matlab's [MATLAB, 2014] built in function using the *recatti* MEEEDEL

3.4 Software

All software was run on the arduino microcontroller and all code were written in the arduino enviroment. The arduino integrated development i (IDE) use the extension .ino butt the code is written in a combination of c and c++. The open source libraries used all were written in this environment so no fixes had to be made.

3.5 Inertial navigation system

3.5.1 Inertial measurement unit

The sensor used was the 9-DOF MPU-9150 from Sparkfun. As sensitivity was preferred and considering that the cube does not rotate around the pivot point at high speeds the gyroscope range was set to 250 deg/s and keeping a high sensitivity of 131 LSB's/dps

3.5.2 Kalman implementation

For the implementation of Kalman filter a library made by Lauszus was used, which is available a the Github repository[Lauszus, 2012]. For an extended explanation of the Kalman implementation, see appendix A.1.2.

Figure 3.3 shows a comparison of the estimated angle at a stationary position using the gyroscope, accelerometer and the Kalman filtered signals separately. The estimated angle using only the gyroscope can be seen by the negative angle ramp. This is due to the bias that is being integrated. The accelerometer readings show that the signal contains a lot of white noise but still has a mean value of zero. The Kalman filtered angle starts following the gyroscope readings but are then corrected when weighed against the accelerometer measurements which can be seen early in the plot. This indicates that the Kalman filter works as it should but the estimated state should still not be considered *true* as the reference is zero.

Figure 3.4 illustrates the estimated angle during a 90 degree turn done by hand. Subfigure 3.4b shows a zoomed view of the plot where the gyroscope bias is obvious but the Kalman angle still follows the true value **Säger jag emot mig sj ? ,...** .

3.5.3 Measurement and process noise

For the Kalman filter to properly work it is essential to know how reliable the process and measurement inputs are. A way of determining the process noise and measurement noise of the IMU is the Allan variance method. The theory of Allan variance is outside the scope of this thesis but for reader reference is a time domain

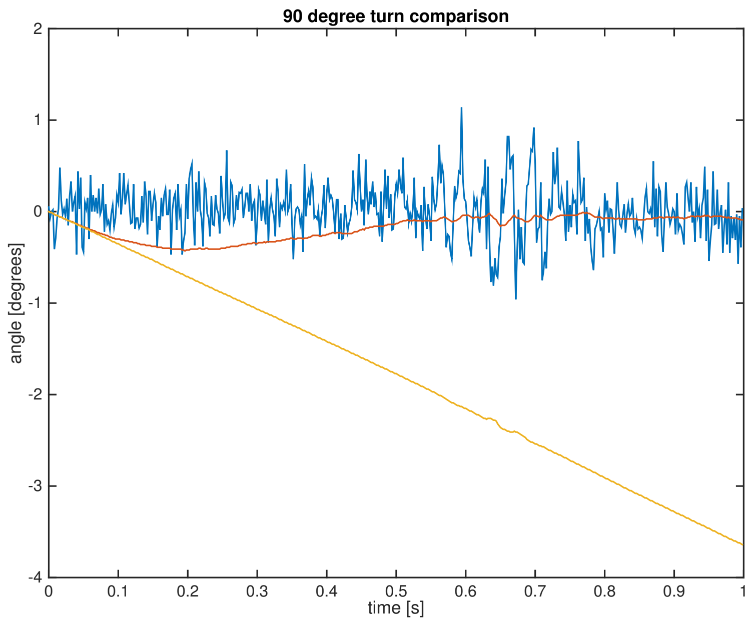
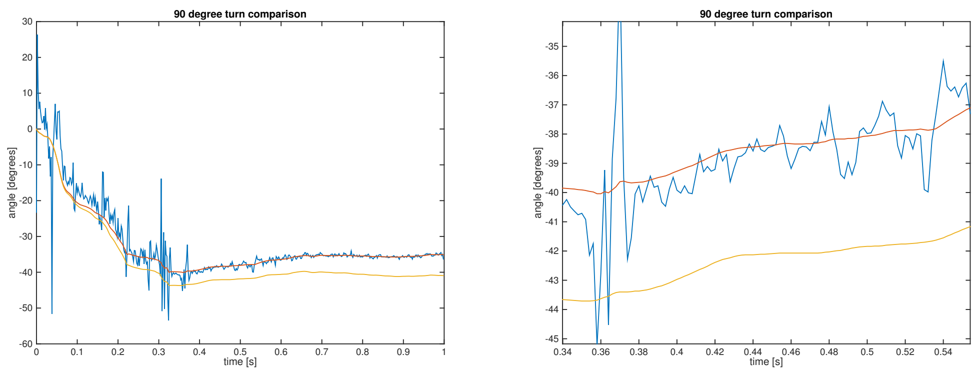


Figure 3.3: Comparison of Kalman filtered signal and original signal (TO BE UPDATED)



(a) 90 degree turn

(b) Zoomed view

Figure 3.4: A figure with two subfigures

analysis technique commonly used to determine the characteristics of errors for inertial sensors[Knight et al., 2009]. The gyroscope data is treated as an external input to the system, so the error and bias from the readings are characterised as process noise. This is then compared to the measurement, the accelerometer, which contains a measurement noise. By gathering samples from the gyroscope and accelerometer at a stationary state the Allan variance can be calculated. The Allan variance is then used to determine the noise and stability of the system. The interesting components of the variance for the IMU are the ARW, VRV and bias stability mentioned in section 2.1. Data from the gyroscope were collected during twelve hours to **achieve, ordval?** a reliable estimate of the bias. The root Allan variance was calculated and is shown in Figure 3.5 where the variance is a function of averaging time τ

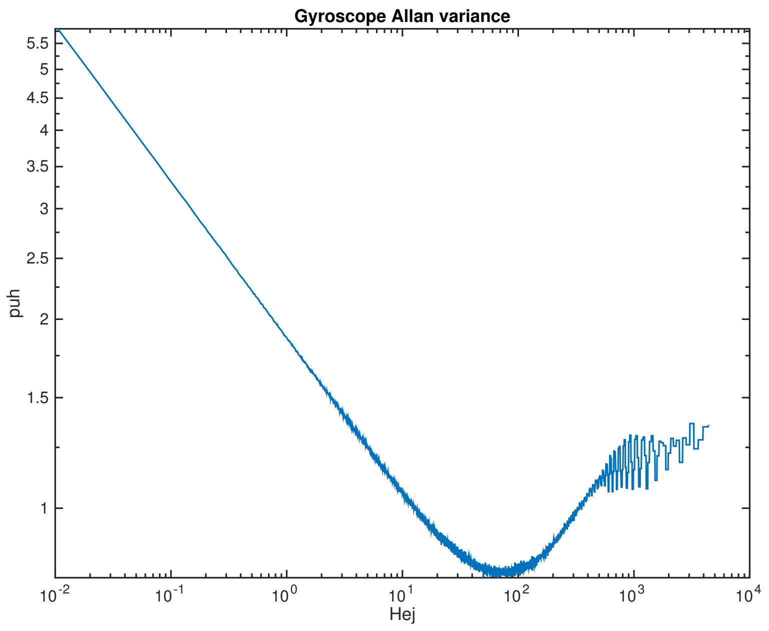


Figure 3.5: Gyroscope root Allan variance FIX PIC

The angle (or velocity, concerning the accelerometer) random walk can be depicted from the plot at an angle of -0.5 while the bias instability is determined by looking at the lowest point of the plot. Allan variance for the accelerometer was **framtagen** in a similar manner. The results can be seen in table 3.1.

Noise term	Degrees/second
Angle random walk	ARW res
Bias stability	BS res
Velocity random walk	VRW res

Table 3.1: Allan variance results

These values corresponds to the process and measurement noise covariance matrices used by the Kalman filter. **Dessa finns bara i appendix ??**

3.6 Control implementation ish

3.6.1 System Control

Allt detta är skrivet för state-space behöver revideras The choosen control method where state space. The problem in to linareise and discretise with good enough precition.The poles for the control system were choosen for the openloop system and the system was simulated in simulink to assure that the respose was as expected. The gain matrix is calculated as the eigenvalues of A-BK = p, (p is the choosen poles), using matlab place It uses a robust calculation algorithm formulated by Kautskyn for determining the gain matrix.

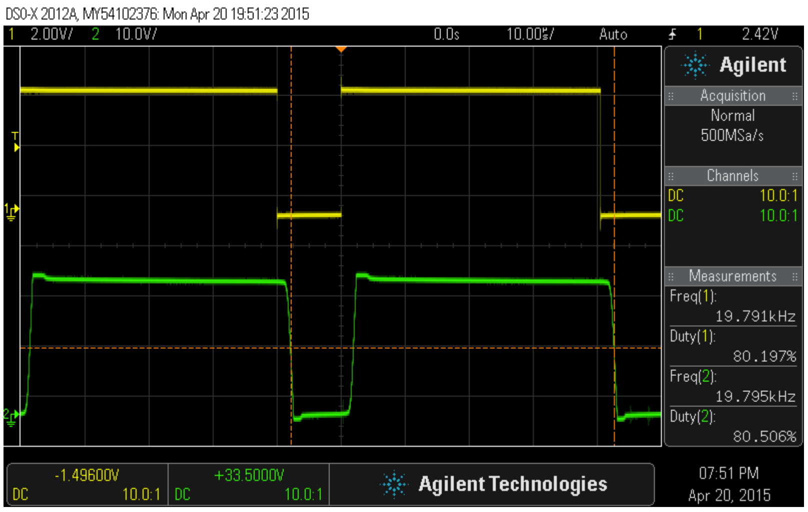
To be able to use a microcontroler the controlsystem must be discrerte. When translating a analytical system to a discrete system a few thing must be considerd. The method of samplig is crutial. If the sampling frequancy is to low the system responce might not detct changes att all or aliasing can cause it to behave unexpectedly. Due to the Nyquist critera the sampling frequency mus be: f>=2*t, twice as high as the smalles frequency you want to be able to detect. Then the signal must be low pass filterd with a cut off frequecny close to t to make sure that no unwanted frequencies is taken into the regulator. All non measured derivatives must be calculated. Simulink does this by calculatin an combination of tustin and backward Euler[mathworks][bilaga]. This causes a numerical error depending on the step in time. And the time step in turn depends on the sampling frequency, too low frequency causes a big stem in time which leads to large numerical errors. Most variables and states will in the microprocessor be handels as double or also know an Double-point floating-point format. This causes an truncation error in every calculation. The double is only the size of a float, 32-bit(?). This means that we only reliably can use 6 decimal digits of precision[float].

3.7 Electronics

3.7.1 PWM

The PWM frequency was set to 20KHz, the maximum the motor controller can cope with [Pololu, 2014]. The duty cycle was varied during the program and calculated from the control system. A typical PWM signal, with 80 % duty cycle, can be seen as the yellow signal in Figure 3.6. The green line is the signal applied the the motor.

A more in depth explanation of the arduino PWM implementation can be found in the appedix A.2.



```
DSO-X 2012A, MY54102376: Mon Apr 20 19:51:23 2015
1  2.00V/   2  10.0V/          0.0s    10.00µs/    Auto    1    2.42V

Acquisition
  Normal
  500MSa/s
Channels
  DC    10.0:1
  DC    10.0:1
Measurements
  Freq(1):
     19.791kHz
  Duty(1):
     80.197%
  Freq(2):
     19.795kHz
  Duty(2):
     80.506%

-1.49600V          +33.5000V                                07:51 PM
DC     10.0:1      DC     10.0:1    Agilent Technologies    Apr 20, 2015
```

Figure 3.6: PWM pic

Chapter 4

Results

Beskriv resultatet.

Chapter 5

Discussion and conclusions

I detta kapitel diskuteras och sammanfattas de resultat som presenterats i föregående kapitel. Sammanfattningen baseras på en resultatanalys och syftar till att svara på den fråga eller de frågor som formuleras i kapitel i.

Förslag på saker som kan vara vettigt PID parameters motor control limitstion hur keffa vi var-fstenbek

The IMU measures very small voltage alterations when the capacitance varies in the mechanical system. Placing such a sensitive device near a motor which induces a flow of bla bla bla

5.1 Discussion

In our case no positions were stable, problabne not because of the sensor placement but due to the control system as a whole. Still it is clear from the result that sensor placement is important and that some positions clearly are better than other

5.2 Conclusions

Successful victory

Chapter 6

Recommendations and future work

6.1 Recommendations

A more extensive research with non-linear control systems has been done at ETH, with the name Cubli,[Gajamohan et al., 2013]

6.2 Future work

An extension of the project would be balancing the cube not only on it's edge but it's corner. To achieve this multiple reaction wheels must be used and a more complicated control system due to changes in moment of inertia caused by angular velocities in the other reaction wheels.

Bibliography

[Andrejasic, 2008] Andrejasic, M. (2008). Mems accelerometers. Technical report, Faculty for mathematics and physics, University of Ljubljana.

[Block et al., 2007] Block, D. J., Åström, K. J., and Spong, M. W. (2007). *The Reaction Wheel Pendulum*. Morgan & Claypool Publishers.

[Chauveau et al., 2005] Chauveau, G., Chazal, D., Nakayama, D., Olsen, E., and Palm, S. (2005). Controlling the reaction wheel pendulum. Technical report, School of electrial engineering , KTH.

[Gajamohan et al., 2013] Gajamohan, M., Muehlebach, M., Widmer, T., and D'Andrea, R. (2013). The cubli: A reaction wheel based 3d inverted pendulum. In *proc. European Control Conference*, pages 268–274.

[Glad and Ljung, 2006] Glad, T. and Ljung, L. (2006). *Grundläggande teori*. Studentlitterarur.

[Johansson, 2003] Johansson, H. (2003). *Elektroteknik*. Institutionen för Maskinkonstruktion, KTH.

[Knight et al., 2009] Knight, N., Han, S., and Wang, J. (2009). Using allan variance to determine the calibration model of inertial sensors for gps/ins integration. Technical report, School of Optoelectronics Science and Engineering, National University of Defense Technology, China.

[Lauszus, 2012] Lauszus, K. (2012). Kalmanfilter. `https://github.com/TKJElectronics/KalmanFilter`.

[MATLAB, 2014] MATLAB (2014). *version 8.4.0 (R2014b)*. The MathWorks Inc., Natick, Massachusetts.

[Pololu, 2014] Pololu (2014). Mc33926. Available from: `https://www.pololu.com/file/0J233/MC33926.pdf`.

[Shiau et al., 2012] Shiau, J.-K., Huang, C.-X., and Chang, M.-Y. (2012). Noise characteristics of mems gyro's null drift and temperature compensation. *Journal of Applied Science and Engineering*, 15(3):239–246.

[Simon, 2001] Simon, D. (2001). Kalman filtering. *Embedded System Programming*, 1(6):72–79.

[Sjöstedt and Ramm, 2015] Sjöstedt, M. and Ramm, A. (2015). Balancing cube. `https://github.com/Axram/CubeProject`.

[Stockwell, 1997] Stockwell, D. W. (1997). Angle random walk. Technical report.

[Welch and Bishop, 2006] Welch, G. and Bishop, G. (2006). Introduction to the kalman filter. Technical report, Department of Computer Science, University of North Carolina at Chapel Hill.

[Woodman, 2007] Woodman, O. J. (2007). An introduction to intertial navigation. Technical report, Computer Laboratory, University of Cambridge.

Appendix A

Additional information

A.1 Elaborative Kalman explanation (TITLE?)

A.1.1 Mathematical background

To understand this recursive filter which use old and new values a *a priori* and *a posteriori* state is defined

$$\hat{x}_k^- \tag{A.1}$$

$$\hat{x}_k \tag{A.2}$$

The *a priori* state in equation (A.1) is defined as the estimate of the current state at the time k. The *a posteriori* state (A.2) is the new estimated state. For the Kalman filter to work properly some criteria has to be fulfilled. The average value of the measurement noise z and process noise w earlier mentioned in section 2.2 has to be zero, i.e. a Gaussian error. z and w also has to be independent of each other. The noise and error in an IMU and many other devices have the characteristics of Gaussian noise.

During the *predict* phase the filter estimates the states using the inputs from the process, i.e the gyroscope in this case. The filter then moves on to the *update* phase where it compares the state to the actual measurement, the accelerometer, see Figure 2.4.

Consider the true state equation (2.2), the Kalman filter first of all estimates a state by neglecting the process noise

$$\hat{x}_k^- = A\hat{x}_{\text{k-1}} + Bu_{\text{k-1}} \tag{A.3}$$

As stated above the Kalman filter uses readings from both the gyroscope and accelerometer to estimate a position closer to the true value. To determine how reliable the process and measurement readings are a noise covariance is defined as

$$Q = E(w_k {w_k}^{\mathrm{T}}) \tag{A.4}$$

$$R = E(v_k {v_k}^{\mathrm{T}}) \tag{A.5}$$

How to determine these covariances are further investigated in section 3.5.3. From here the *a priori* error covariance matrix is introduced to symbolize the noise in the process measurement

$$P_k^- = AP_{\text{k-1}}A^T + Q_k \tag{A.6}$$

During the *update* phase the accelerometer measures are used. The measurement *innovation* is calculated as

$$\tilde{y} = z_k - H\hat{x}_k^- \tag{A.7}$$

The *innovation* is a residual that reflects the relation between the predicted measurement and the actual measurement. A measurement *innovation* of zero indicates a perfect agreement. The measurement *innovation* covariance is calculated as

$$S_k = HP_k^- H^T + R \tag{A.8}$$

The *innovation* covariance is very similiar to the *priori* error covariance but represents the measurement instead. From here the core of the Kalman filter can be calculated, the Kalman gain

$$K_k = P_k^- H^T S^{\text{-1}}{}_k \tag{A.9}$$

This indicates how reliable the measurement is. Note that if the measurement covariance error (A.5) is large, the Kalman gain will be small and vice versa if the *a priori* error covariance is large. By now the *a posteriori* state can be estimated by

$$\hat{x}_k = \hat{x}_k^- + K_k\tilde{y}_k \tag{A.10}$$

A current state has been estimated, this would be the angle used for the control loop. Finally the *priori* error is updated to be used for the next *a priori* error.

$$P_k = (I - K_k H)P_k^- \tag{A.11}$$

The filter now returns to the measurement phase seen in figure 2.4.

A.1.2 Implementation

The Kalman filter cycles two states, the *predict* and *update* phases. The filter is using discretized steps making the filter implemented on a microprocessor **fairly simple**. An implementation of the Kalman filter on the IMU would look something like this

Consider the equation (A.3) from chapter A.1 . At first the filter predicts the state

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{x}_{\text{k-1}} + \mathbf{B}u_{\text{k-1}} \tag{A.12}$$

Where **FETSTILTA MATRISNOATIONER**

$$\mathbf{x}_k = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k u_{\text{k-1}} = \dot{\theta} \tag{A.13}$$

$$\mathbf{A} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \tag{A.14}$$

The states used here are the angle as well as the bias. Next up is the *priori* error covariance

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{\text{k-1}}\mathbf{A}^T + \mathbf{Q}_k \tag{A.15}$$

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^-_{\text{k}} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}_{\text{k-1}} \begin{bmatrix} 1 & 0 \\ -\Delta t & 1 \end{bmatrix} + \begin{bmatrix} Q_{\,\theta} & 0 \\ 0 & Q_{\dot{\theta}_b} \end{bmatrix} \Delta t \tag{A.16}$$

The innovation covariance mentioned in A.8

$$\mathbf{S}_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^-_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} + R \tag{A.17}$$

Kalman gain

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix}_k = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^-_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} S_k^{-1} \tag{A.18}$$

The predicted states are then updated with the weighed measures from the accelerometer

$$\begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}_k = \begin{bmatrix} \theta \\ \dot{\theta}_b \end{bmatrix}^-_k + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}_k \tilde{y} \tag{A.19}$$

And then at last the error covariance

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}_{\text{k}} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}_k \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^-_k \tag{A.20}$$

This is done during each loop of the microcontroller to ensure that the estimated state is as near the true state as possible.

A.2 Arduino PWM implementation

The Arduino microcontroller has a couple of different standard *pulse width modulation* (PWM) modes. The standard frequencies are 490 and 980 Hz. Using a significantly higher frequency has many advantages, especially for the control system. The chosen motor controller could work in two modes, one of them being ultrasonic the other below ultrasonic. Setting the PWM frequency above the arduino standard frequency is preferably because the the motor becomes more efficient **SOURCE** and the human ear can not hear frequencies above $\tilde{2}0$ KHz. The Arduino Uno's maximum PWM frequency is $\tilde{6}1$ KHz, when set to high the motor controller consumes power so good setting is just above hearing range. WHICH IS TOO MUCH

FOR THE MOTOR CONTROLLER WHY. A bit of register manipulation had to be done to accomplish an acceptable PWM frequency.

The Arduino has 3 clocks that controls various pins and functions in the environment. The clocks, and their respective pins and bit size, are:

Timer 0 controls digital pin 10 and works in 8-bits it also used as reference for the delay(), millis() and micros() built in functions. i.e. changing timer 0 could disturb the code if it is using these functions.

Timer 1 The 16-bit timer. Controlls pin 5 and 6 and has more modes of operation than the other two timers

Timer 2 another 8-bit timer responsible for pin 3 and 11 on the Arduino Uno.

Each of these timers can be set the various PWM modes individually, —more on this later—. All timers have prescalers which divides the chips clock speed to vary the frequency of the PWM signal in rough discrete steps (the difference between the fastest and the next to fastest fequency on timer1 is 53 KHz)(see avr documentation for mor info) **källa h'r eller ??**.

When using PWM, the PWM duty cycle is used (and optionally frequency). Once the PWM duty cycle is set this works parallel to the main program and does not effect the main program but is rather run **annan tråd??**(except when changing the PWM parameters). For example this is useful when only one Arduino is used for multiple tasks, and computing time becomes expensive. A non inverted mode where used, meaning that 100% duty cycle equals full power all the time. There are multiple modes of PWM on the Arduino. The main ones that can be choose from are fast - and phase corrected PWM. When using the phase correct PWM (pcPWM) a counter counts up every clock cycle until the counter reaches a compare value, *TOP*. When TOP is reached the counter is reset or "set" according to the AVR documentation, be aware that for the method to work, the counter has to be adjusted to "set" not "clear" or "set bottom". Then the duty cycle has to be set to a value between 0 and TOP. Choosing duty to TOP/2 will yield a 50% duty cycle. For modes where the TOP value is not specified it defaults to 0xFF or 255 in decimal.

When using the pcPWM (on timer2 in this case), mode 5, the TOP value and duty cycle for ONE of timer2's pin's are the same variable. This means that one can not change the duty cycle for a certain frequency. The frequency can then be explained by equation A.21

$$frequency = \frac{clock}{2 \cdot prescaler \cdot TOP} \tag{A.21}$$

But the other pin (nr 3) is left uneffected. So if the desired PWM is 20 KHz it is done by choosing 100 as TOP (OCR2A for timer2) and the prescaler value to 8 (TCCR2B = (1«CS21) pin 11 will always have 50% duty cycle at 20 KHz. But the duty cycle for pin 3 can be set continiously. you can choose another duty cycle for pin 3 **HÄR ÄR DET KNAS ?** (OCR2B) while maintaining 10 KHZ due to the TOP, OCR2A value is not changed. The disadvantage of this method is that it requires 2 pins, while only one is used. A minor price to pay for a specific frequency and reduced headache from the high pitched noise.

To choose mode you must set the correct WGM vaues in the TCCR2A and TCCR2B registers. The fourth letter "2" represents timer 2. Timer 2 is 8-bit so it has a maximum of 8 modes. If instead timer 1 is used more modes can be chosen allowing for a greater amount of operation modes. Other Arduino boards might have a greater number of timers which allows for more than three different PWM's, with non default frequencies, being used at the same time.

Appendix B

Proofs

B.1 System dynamics

The first equations of motion for the system is derived by using Lagrangian dynamics Firstly by expressing the generalized coordinates, forces, the energy functions and Lagrangian. And then acquire the equations of motion from the Lagrange equation [Block et al., 2007]. Consider the Lagrangian equation

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \left(\frac{\partial \mathcal{L}}{\partial q_i}\right) = \tau_i \tag{B.1}$$

Where τ is the generilzed force, in this case a torque. The q operator is the generalized coordinate which now is an angle. The cube's angular momentum is counteracted by the flywheel and the system can be divided into two parts, One considering the movement of the cube, the other the flywheel.

$$\tau_k = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \left(\frac{\partial \mathcal{L}}{\partial \theta}\right) \tag{B.2}$$

$$-\tau_k = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) - \left(\frac{\partial \mathcal{L}}{\partial \phi}\right) \tag{B.3}$$

Whereas θ represents the angle of the cube and ϕ is the position of the flywheel. The Lagrange equation is derived from the difference in kinetic energy and potential energy of the cube

$$\mathcal{L} = E_k - E_p \tag{B.4}$$

$$E_k = \frac{I_c \cdot \dot{\theta}^2}{2} + \frac{I_f \cdot \dot{\phi}^2}{2} \tag{B.5}$$

$$E_p = M_{tot} \cdot g \cdot l \cdot \cos\theta \tag{B.6}$$

The lagrangian (B.4) is then

$$\mathcal{L} = \frac{I_c \cdot \dot{\theta}^2}{2} + \frac{I_f \cdot \dot{\phi}^2}{2} - M_{tot} \cdot g \cdot l \cdot \cos\theta \tag{B.7}$$

The kinetic energy depends on the angular velocities of the main construction as well as the flywheel fixed to the motor. Note that the total moment of inertia I_c is defined around the pivot point of the cube. The potential energy has been defined as being at its maximum when the cube is balancing in an upright position. The construction is considered to be symmetric and hence the gravitational force is applied on the center of the cube. Equation (B.2) and (B.3) with (B.4)

$$I_c \cdot \ddot{\theta} - M_c \cdot g \cdot l \cdot \sin\theta = -\tau_k \tag{B.8}$$

$$I_s \cdot \ddot{\phi} = \tau_k \tag{B.9}$$

Equation B.8 together with B.2 express how the flywheel and main construction depend of each other

$$I_c \cdot \ddot{\theta} - M_c \cdot g \cdot l \cdot \sin\theta = -I_s \cdot \ddot{\phi} \tag{B.10}$$

To use linear control methods the model has to be linearised. This is done at the instable equilibrium where the cube is balancing. Consider the sinus term at the equilibrium point where θ equals 0. The term can then be expressed with Maclaurin expansion

$$sin\theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} ... \approx \theta \tag{B.11}$$

In the Laplace domain the flywheel angle ϕ can then be described by a transfer function and θ

$$\phi(s) = \frac{M_{tot} \cdot g \cdot l \cdot -I_c \cdot s^2}{I_f \cdot s^2} \theta(s) \tag{B.12}$$

The transfer function that explains of ϕ depends of θ is then

$$G_1(s) = \frac{M_{tot} \cdot g \cdot l \cdot -I_c \cdot s^2}{I_f \cdot s^2} \tag{B.13}$$

As the controller is voltage controlled **ordet controll x2** the angle of the cube has to be expressed an input voltage. The torque exerted on the flywheel is described **ordvalen nu lol** by the current flowing through the motor as well as the motor and gear efficiency

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \cdot i \tag{B.14}$$

The voltage across the two poles of the motor can be described by the current flowing through the motor

$$U = L\frac{di}{dt} + iR + \dot{\phi}K_e \tag{B.15}$$

Equation B.15 and B.14 gives

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \cdot \frac{U - E_{\text{emf}}}{R_m} \tag{B.16}$$

Note that the motor inductance in neglected in equation (B.16), that is due to the inductance time constant which is fast considering the rest of the system and is not vital for the control system [Chauveau et al., 2005]. The induced voltage can be described as a function of motor speed.

$$E_{\text{emf}} = K_{\text{emf}} \cdot \dot{\phi}_r \tag{B.17}$$

By moving to the Laplace domain and combining equations B.16, B.17, B.12 and B.13 one gets

$$I_f \cdot \eta_m \cdot \eta_g \cdot \Gamma \cdot G_1(s) \cdot s^2 + \frac{K_t \cdot K_e \cdot s}{R_a}\theta(s) = \frac{Kt}{R}U(s) \tag{B.18}$$

And at last the cube's angle in the Laplace domain can be described by a transfer function and the voltage input U

$$\theta(s) = \frac{K_t \cdot I_f \cdot \eta_m \cdot \eta_g \cdot \Gamma \cdot s^2}{R_a(M_{tot} \cdot g \cdot l - I_c \cdot s^2)(I_f \cdot s^2 + \frac{K_t \cdot K_e \cdot s}{R_a})}U(s) \tag{B.19}$$