



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

Reaction wheel balanced robot

DESIGN AND SENSOR ANALYSIS OF INVERTED PENDULUM ROBOT

ALEXANDER RAMM & MIKAEL SJÖSTEDT

KTH ROYAL INSTITUTE OF TECHNOLOGY

INDUSTRIAL ENGINEERING AND MANAGEMENT

Reaction wheel balanced robot

Design and sensor analysis of inverted pendulum robot

MIKAEL SJÖSTEDT
ALEXANDER RAMM

Bachelor's Thesis in Mechatronics

Supervisor: Daniel Frede

Examiner: Martin Edin Grimheden

Approved: 2015-05-20

TRITA MMK 2015:18 MDAB071

Abstract

Some robots can do remarkable things. Balancing something unstable is one of them. This paper covers the design of such a robot and evaluates how the sensor is affected by its position and in turn the robots ability to balance. A robot with a reaction wheel to maintain balance was built and the sensor data was investigated.

From the result one could see that the sensor placement was of importance. The best placement was the one closest to the center of the cube, far away from the reaction wheel.

Sammanfattning

Design och sensorutvärdering av inverterad pendel-robot

En del robotar kan utföra förbluffande manövrar. Att balansera något instabilt är en av dem. Den här rapporten täcker konstruktionen av en sådan robot och undersöker hur sensorn påverkas av sin position och i sin tur hur balansförmågan påverkas av sensorn. En robot som använde ett svänghjul som hjälp för att balansera byggdes och sensordatan undersöktes.

Från resultatet visade att sensorplaceringen var viktig. Den bästa placeringen var den närmast den robotens centrum.

Preface

We would like to thank Daniel Frede, our supervisor, for his continuous support, pinpointing our faults with *German precision*.

Staffan Qvarnström and Tomas Östberg for their mechanical and practical expertise and counselling.

We would also like to thank our fellow student colleagues David Tennander & Viktor Gjurovski, who has been carrying out an equivalent project, for rewarding discussions and ideas.

Furthermore Filip Stenbeck and Aron Nygren for their inspiration and enthusiasm. And definitely all the student assistants who have helped us throughout the course.

Alexander Ramm
Mikael Sjöstedt
KTH, may, 2015

Contents

Abstract	iii
Sammanfattning	v
Preface	vii
Contents	ix
Nomenclature	xi
1 Introduction	1
1.1 Background	1
1.2 Purpose	2
1.3 Scope	3
1.4 Method	3
2 Theory	5
2.1 Inertial Measurement Unit	5
2.1.1 Background	5
2.1.2 Accelerometer	6
2.1.3 Gyroscope	7
2.2 Kalman filter	8
2.2.1 Introduction	8
2.2.2 State Estimator	9
2.3 Model dynamics	10
3 Design and implementation	13
3.1 Problem Formulation	13
3.2 Design	14
3.2.1 Frame	14
3.2.2 Arduino	15
3.2.3 Motor controller	15
3.2.4 Motor	15
3.3 System analysis	15

3.4	Software	15
3.5	Inertial navigation system	16
3.5.1	Inertial measurement unit	16
3.5.2	Kalman implementation	16
3.5.3	Measurement and process noise	18
3.6	Electronics	19
3.6.1	PWM	19
4	Results	21
5	Discussion and conclusions	23
5.1	Discussion	23
5.2	Conclusions	24
6	Recommendations and future work	25
6.1	Recommendations	25
6.2	Future work	25
	Bibliography	27
	Appendices	
A	Additional information	29
A.1	The Kalman filter	29
A.1.1	Mathematical background	29
A.1.2	Implementation	30
A.2	Arduino PWM implementation	31
B	Proofs	35
B.1	System dynamics	35
C	Schematics	39
C.1	Electrical circuit	39

Nomenclature

Symbols

Symbol	Description
\mathcal{L}	Lagrangian operator
θ	Cube angle (Rad)
ϕ	Flywheel angle (Rad)
q	Lagrange operator
E_k	Kinetic energy (J)
E_p	Potential Energy (J)
I_c	Inertia of the cube (Kg · m ²)
I_f	Inertia of the flywheel (Kg · m ²)
M_{tot}	Total mass of the cube (Kg)
M_f	Mass of the flywheel (Kg)
i	Current (Amp)
K_t	Motor torque constant (Nm / Amp)
E_{emf}	Induced voltage (V)
K_{emf}	Motor voltage constant (V · s/Rad)
U	Voltage across motor poles (V)
R_m	Motor internal resistance (Ohm)
η_m	Motor efficiency
η_g	Gear efficiency
Γ	Gear ratio
z	Measurement noise
w	Process noise

Abbreviations

Abbreviation	Description
CAD	Computer Aided Design
CAE	Computer Aided Engineering
DC	Direct Current
DOF	Degrees of freedom
dps	Degrees per second
IC	Integrated circuit
I ² C	Inter-Integrated circuit
IDE	Integrated Development Interface
LSB	Least significant bit
MCU	Microcontroller
MEMS	Microelectromechanical Systems
PLM	Product Lifecycle Management
PWM	Pulse With Modulation
RMS	Root Mean Square
RPM	Revolutions per minute
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus

Chapter 1

Introduction

“If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.”

Edward A. Murphy Jr.

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 widespread 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 one 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 a one *degree of freedom* (DOF) inverted pendulum type structure using a reaction wheel is no new concept, and became more accessible with the introduction of inexpensive microcontrollers. The use of automated control has been growing at a rapid pace the last decades and is being implemented more and more in consumer related products. The range of available sensors for use in automated control is larger now than ever.

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 a 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 is the authors wish to contribute knowledge about how such a device could be built and evaluate some of the capabilities and restrictions of such a machine.

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

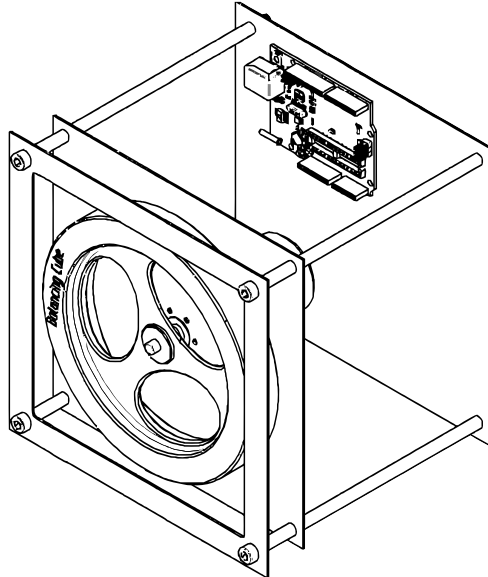


Figure 1.1: Concept design

The dynamics are mostly effected by the control system, which is responsible for accelerating the motor and thus the reaction wheel in the correct angular direction, to maintain balance. The parameters in the control system effect 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 is to implement knowledge in mechanics, electronics and control theory to result in a functioning robot. A research part whose topic was to investigate how the system was effected by sensor placement. This could be concentrated to a question

1.3. SCOPE

What sensor position is least inflicted by noise, when the cube is stationary?

How much does the best and worst position, with regards to the positions determined in the question above, effect the system performance when balancing?

The sensor used for this project are 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 device 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 sensor which position were to be examined was an IMU. Only a few key positions of the sensor was examined.

The effects that were looked at was the ones related to the control system. Only data from the IMU axes used in the specific control system were examined, the placement effect on other axes was not taken into consideration. All control model dynamics was done non-discretely.

1.4 Method

At first, data from the IMU was collected at different positions during a stationary state where the frame was rigidly mounted. The motor was accelerated with a sinusoidal current to simulate the same conditions as during a balancing state. The IMU was then moved to a position closer to the motor with the hypothesis that the motor induce an electrical field that effects the measurement unit. Six recordings were made for each placement and the variance was calculated thereafter.

Five different positions were examined. They can be seen in Figure 1.2. Six measurements were taken for every position. The expected value and standard deviation for the particular position was calculated.

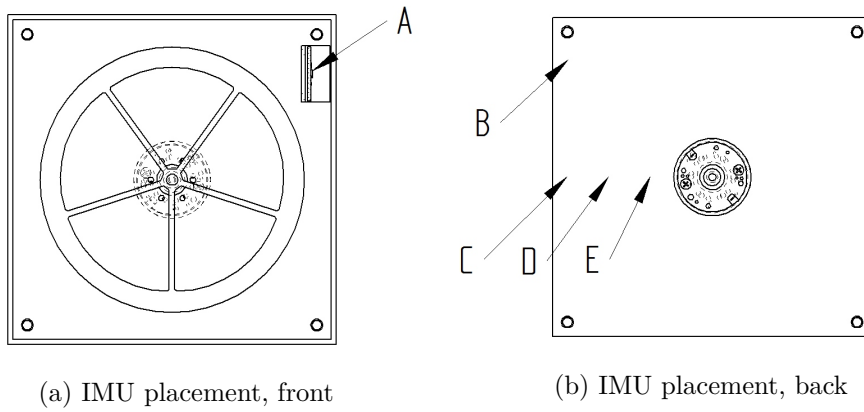


Figure 1.2: IMU placements

Chapter 2

Theory

"Nothing is as easy as it looks."

Murphy's law

This chapter covers some of the theory that is required if one would 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 discusses 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 regulator. 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. An IMU is often rated for several degrees of freedom, a unit specified as 6-DOF usually 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 navigation system a Cartesian coordinate system is defined in Figure 2.1

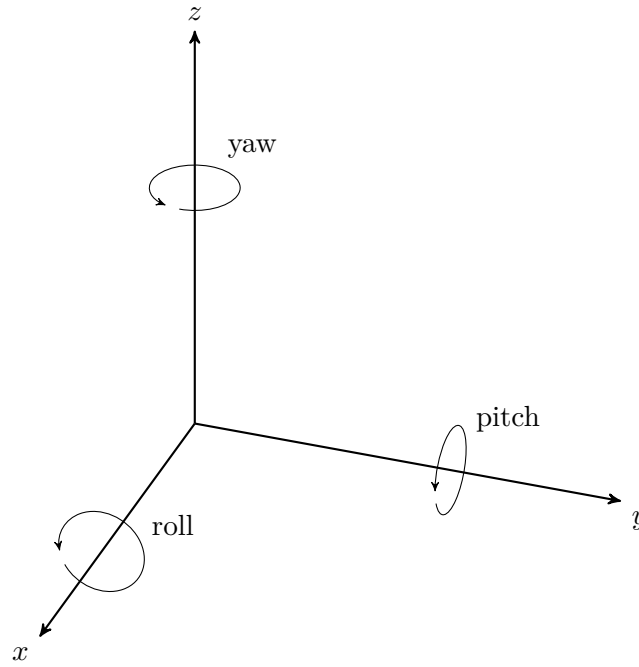


Figure 2.1: Cartesian coordinate system

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 [Andrejasic, 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 of 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,

2.1. INERTIAL MEASUREMENT UNIT

seen in figure 2.2. The capacitance is thereafter used to determine the displacement of the mass and then calculate the force applied to it with the known spring mass system.

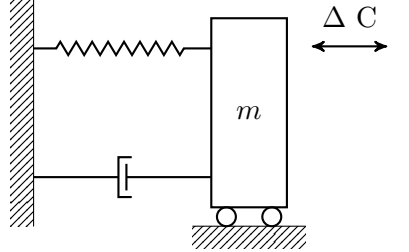


Figure 2.2: Accelerometer capacitance

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.

To specify how much the noise effects an angle estimation made by the accelerometer, a *velocity random walk* (VRW) is introduced [Stockwell, 1997]. The noise from the accelerometer during a stationary position can be characterized as a white noise, and an integrated value is "walking" 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

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

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

A micromechanical gyroscope is, as the accelerometer, effected by a bias. This is often due to friction caused by moving parts or production variations that 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

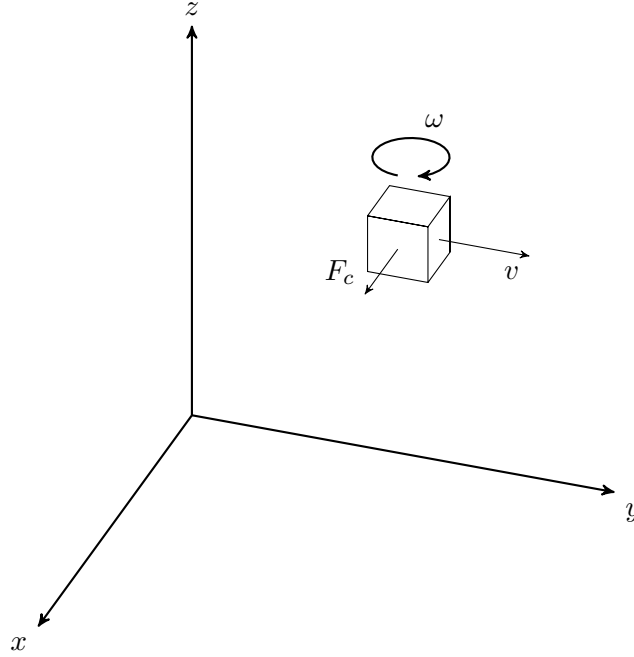


Figure 2.3: Gyroscope pic

electronics [Woodman, 2007]. Thus a *bias stability* is introduced as an indication of how the bias may change during a period of time.

Other 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.

The concepts of ARW, VRW and bias stability that has been introduced are an indication of how precise the output from 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 is accurate for short periods, but over time the estimated position *drifts* [Shiau et al., 2012]. How much the position drifts is correlated to the ARW, VRW and bias stability. 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 calculation.

By using a Kalman filter the drift can effectively be minimized. If the measurements from both the gyroscope and accelerometer are considered, and with some

2.2. KALMAN FILTER

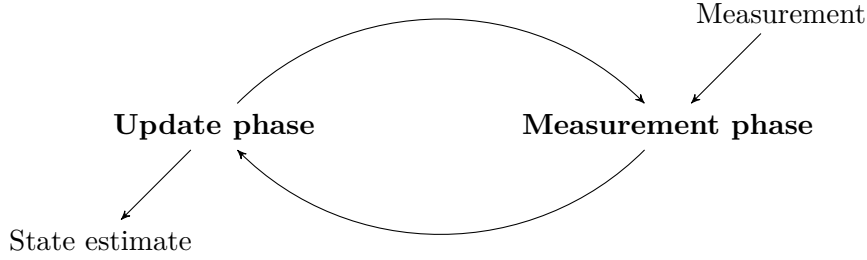


Figure 2.4: Kalman phases

help of probability theory the estimated angle 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 to calculate an estimation of the current state.

One could say that the Kalman filter moves between two phases, see Figure 2.4. First of all it estimates a state using a known process. 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 mentioned above, a state based estimator. By using the last state and the one before that it can derive a better estimate of the current state. Consider the true state and the measured value at a time k

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (2.2)$$

$$z_k = Hx_k + v_k \quad (2.3)$$

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 estimate a reliable state \hat{x}_k

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (2.4)$$

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 help create a control system, the physical model has to be translated to a mathematical model. The system can be estimated much like a one-degree-of-freedom inverted pendulum model [Chauveau et al., 2005]. See Figure 2.5 of how the model can be simplified.

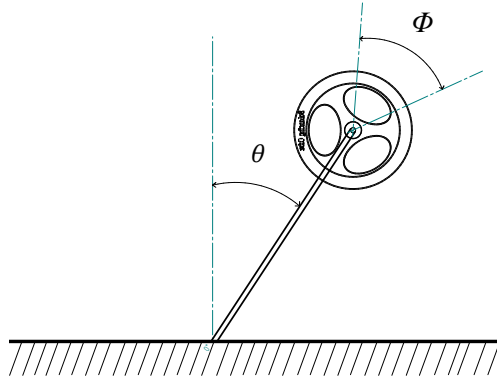


Figure 2.5: Cube modelled as a reaction wheel pendulum

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

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

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

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

2.3. MODEL DYNAMICS

As the controller input is regulated by voltage 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

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \cdot i \quad (2.7)$$

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

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

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

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

With the transfer function from equation 2.9 initial PID-parameters for the demonstrator could be estimated.

Chapter 3

Design and implementation

"It is impossible to make anything foolproof, because fools are so ingenious."

Murphy's law

This part covers the design of the demonstrator and the motivation of components. 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 an engineering problem. The cube should be a robot that, using a reaction wheel can balance on its edge. All components were mounted in the robot, then only requiring a power source.

The main components consisted of a frame, reaction wheel, motor, motor controller, Arduino, and an IMU.

The 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 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 thus a more powerful motor which in turn requires another controller.

The frame of the cube was constructed in aluminium consisting of 3 plates. By keeping the heavy parts symmetric considering the electrical components, the subsystems could be placed arbitrarily and the center of gravity still 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.

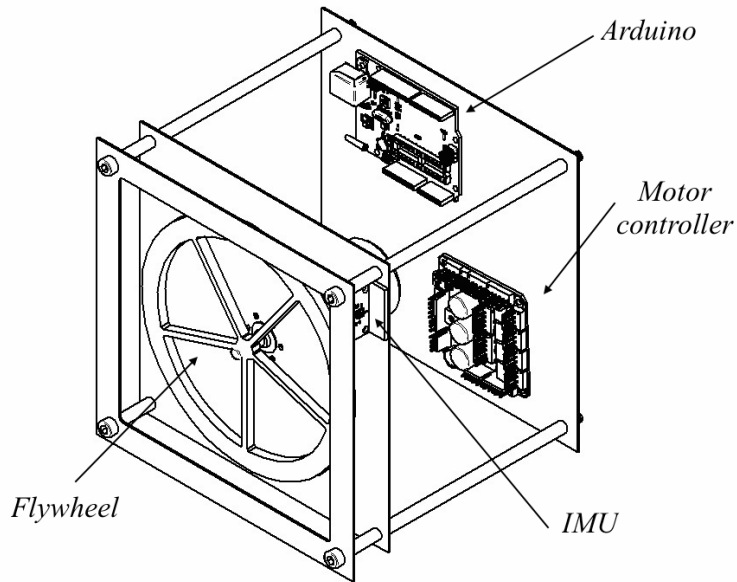


Figure 3.1: The final design of the cube

3.3. SYSTEM ANALYSIS

3.2.2 Arduino

The Arduino is an 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.

3.2.3 Motor controller

A factory made full H-bridge DC motor controller on a chip was used. The main criteria the controller had to meet was current and voltage capabilities suitable for the motor. A motor controller from Pololu was used specified at 28V and able to handle continuous currents of three ampere. The chosen controller also had a safety function that disabled the motor if an over-current ran through the controller.

The motor controller 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 appendix C.1

3.2.4 Motor

To apply torque to the flywheel, a Maxon DC motor with a nominal voltage of 24V was used. The torque exerted on the flywheel is correlated 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 eq. 2.9 is used

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

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.

Matlab was then used to estimate rough proportions of the PID values. Which are to be fine tuned during experiment.

3.4 Software

All software was run on the Arduino microcontroller and all code were written in the Arduino environment. The Arduino *integrated development* (IDE) use the extension

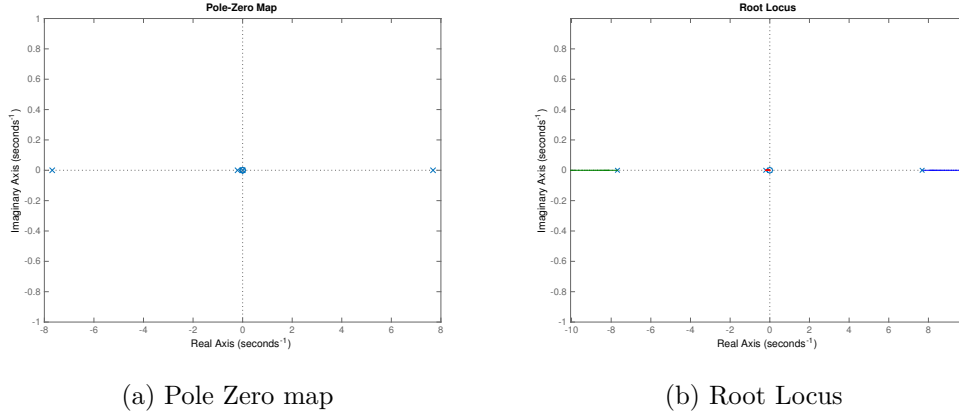


Figure 3.2: A figure with two subfigures

.ino but the code is written in a combination of C and C++.

All other calculations to examine the system and performance figures was made using MATLAB.

3.5 Inertial navigation system

3.5.1 Inertial measurement unit

The sensor used was a 9-DOF MPU-9150 from Sparkfun. As sensitivity was preferred and considering that the cube does not rotate around the pivot point at high velocities the gyroscope range was set to lowest possible 250 deg/s and thus keeping a 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 at 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 as the negative angle ramp. This is due to the bias error 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 converging to the gyroscope measurements but are then corrected when weighed against the accelerometer which can be seen early in the plot. This indicates that the Kalman filter works but the estimated state should still not be considered as *true*.

3.5. INERTIAL NAVIGATION SYSTEM

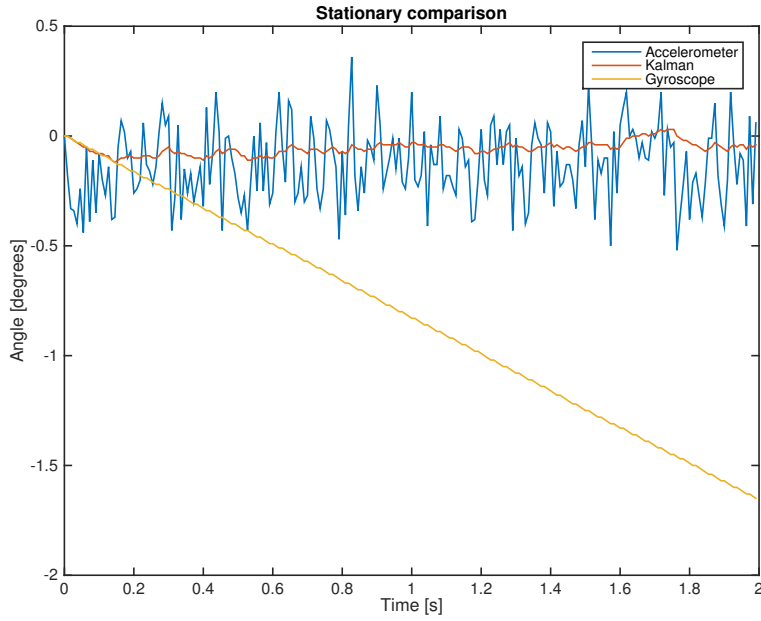


Figure 3.3: Comparison of Kalman filtered-, gyroscope- and accelerometer angle

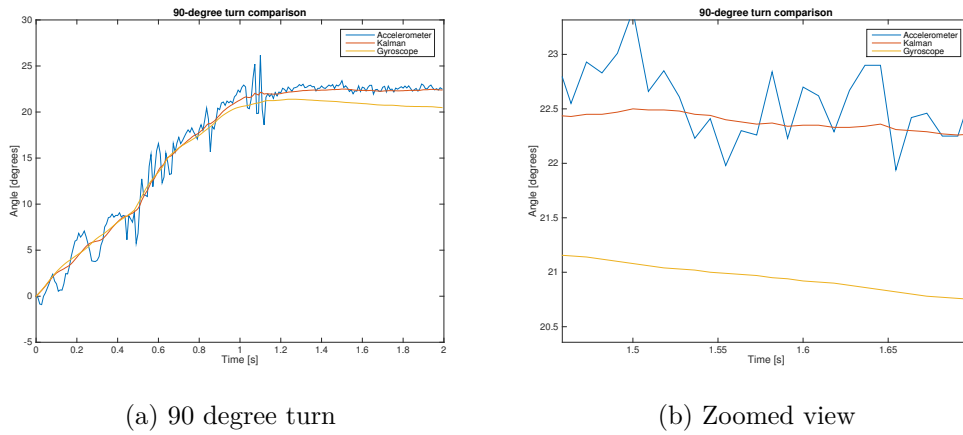


Figure 3.4: Comparison of the estimated angle during a 90-degree turn

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 distinct but the Kalman angle is converging towards the true angle.

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 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 was collected during twelve hours to retrieve 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

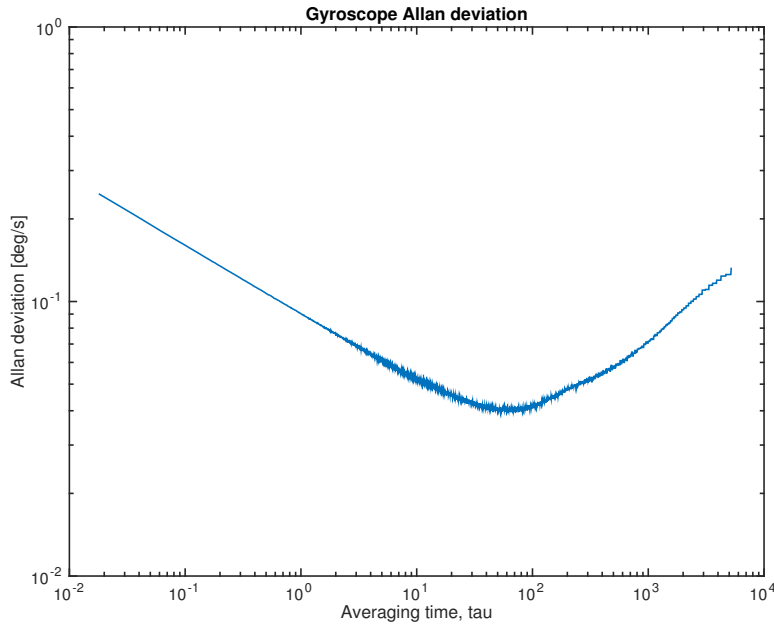


Figure 3.5: Gyroscope root Allan variance

The angle (or velocity, concerning the accelerometer) random walk can be depicted from the plot where the averaging time equals a second while the bias instability is determined by looking at the lowest point of the plot. Allan variance for the accelerometer was retrieved in a similar manner. The results can be seen in

3.6. ELECTRONICS

table 3.1.

Noise term	Degrees/second
Angle random walk	0.00905
Bias stability	0.00395
Velocity random walk	0.014

Table 3.1: Allan variance results

These values corresponds to the process and measurement noise covariance matrices used by the Kalman filter A.4 and A.5.

3.6 Electronics

3.6.1 PWM

Pulse-width modulation (PWM) is a technique used to make a stationary signal 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[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 actual supply voltage. They duty cycle can be changed during operation unlike liner type voltage regulators.

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 can be seen in Figure 3.6 as the upper signal. The lower signal is the one applied to the motor.

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

CHAPTER 3. DESIGN AND IMPLEMENTATION

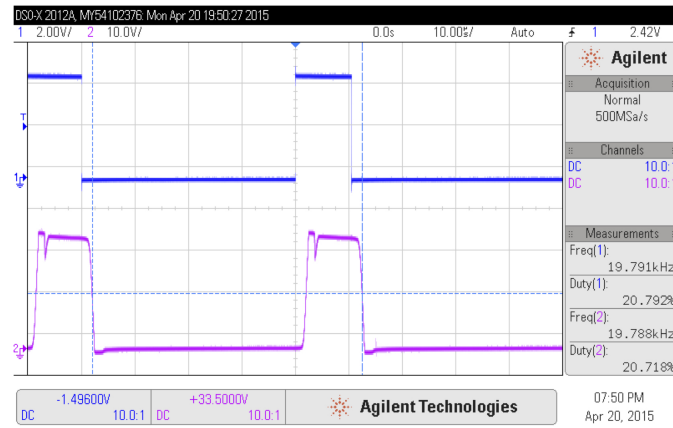


Figure 3.6: PWM, signal from Arduino and applied on motor

Chapter 4

Results

The test, to determine the disturbances in various positions was conducted during a stationary condition. Only moving the IMU to the different positions as seen Figure 4.1. For every position six measurements were made.

The standard deviation and mean value of the IMU signal at the different placements are seen in table 4.1.

Position	Motor	Angle	
		Standard deviation	Mean value
A	Enabled	0.92	44.19
B	Enabled	0.87	44.44
C	Enabled	0.59	44.26
D	Enabled	0.52	44.00
E	Enabled	0.42	44.15
	Disabled	0.11	44.40

Table 4.1: Stationary IMU data results

A comparison of the worst case scenario and the best measurement from position D and E can be seen in Figure 4.2

Furthermore position D and E were planned to be evaluated when balancing, to test system performance for different sensor positions. But since the cube did not balance long enough to render any unambiguous results these tests were not conducted.

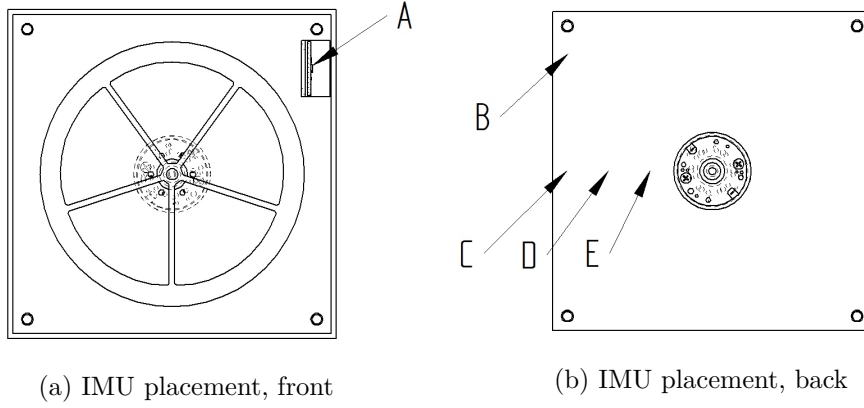


Figure 4.1: IMU placements

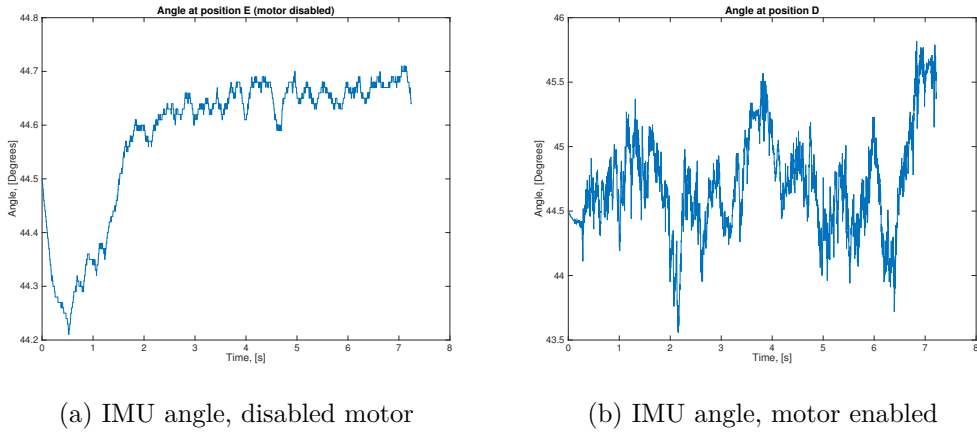


Figure 4.2: IMU angle during stationary test

Chapter 5

Discussion and conclusions

5.1 Discussion

At first the sensor data was tested for different positions as described in section 1.4. The results seen in Table 4.1 and clearly shows that the closer the IMU was placed to the center of the cube, the standard deviation of the signal during a stationary state decreased. From this, the original hypothesis that the motor would induce an electromagnetic disturbance could be discarded.

Rather, for this setup, with a small motor, the quality of the sensor data was more inflicted by the vibrations in the structure. By placing the sensor closer to the center of the structure the standard deviation decreased. The center where the motor is mounted was more rigid and potentially dampens the vibrations.

The variations in the mean value are interesting as well. There are no relationship between the values and their respective positions. No obvious conclusion can be drawn from the mean values. Nor can we measure the correct value or even comment on what it should be, other than close to 45 degrees. Still it is important that the value is close to the real world otherwise the IMU might perceive a displaced balance point and thus having troubles balancing.

The worst position was position A, the one on the front of the cube. At that position the IMU was close to the rotating reaction wheel. The reaction wheel itself is made of steel, a conductive and magnetic metal. This might contribute to the noise experienced by the sensor. This and the fact that it is far from the centre is the reason that it performed worse than all other tested positions.

The cube could not balance and was unstable for every sensor position. Probably not because of the sensor placement but due to the control system and design as a whole. Still it is clear from the result that sensor placement is important and that some positions are better than others.

If a bigger cube was built with a larger inertia the system would be slower and hence not require as fast control system. As for now, the gear used was a spur gear causing a slight delay when switching directions. This is something that implicates

things as the system was not as responsive as required.

5.2 Conclusions

For a construction like the one used in this project with a rather small motor the electromagnetic induction from the motor had a negligible effect on the sensor, thus should the sensor be placed only considering the vibrations in the structure. Also avoid positions close to rotating metal structures.

Since the cube did not balance, a more powerful motor or efficient gearing would have been favourable. Though increasing the size of the motor could make the results invalid. To actually achieve balance a faster system with more torque would have been needed.

Chapter 6

Recommendations and future work

6.1 Recommendations

A larger motor with higher stall torque is better for the task, as long as a suitable motor controller is used. Also an encoder to measure motor RPM is an advantage if a better control system is to be devised. Although a larger motor might induce higher currents which could render the sensor placement results unfavourable.

Also if a more extensive project is desired, a research with non-linear control systems has been done at ETH (Eidgenössische Technische Hochschule Zürich), with the name Cubli[Gajamohan et al., 2013], which was the inspiration of this project.

6.2 Future work

To get a preciser result regarding the IMU position, more parameters should be properly examined. Various motors and motor controllers that allow different amounts of induced currents should also be varied. If the current is measured separately its contribution could be determined independent from the vibration disturbance as well. To investigate the system performance it is of importance to achieve balance for more than a couple of seconds, this might be the main focus.

The control system can be extended by adding features to the robot, such as making it able to jump up from a stationary state. This would require an external brake if not the motor is powerful enough to stop the reaction wheel. Something as a mechanical lock that instantly locks the reaction wheel is the most simple solution but a v-brake with two brake-callipers similar the ones used for bicycles is preferable.

An extension of the project would also 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.
- [Atmel, 2010] Atmel (2010). Atmega48a/pa/88a/pa/168a/pa/328/p. Available from: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf.
- [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 electrical 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*. Studentlitteratur.
- [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>.
- [Pololu, 2014] Pololu (2014). Mc33926. Available from: <https://www.pololu.com/file/0J233/MC33926.pdf>.

BIBLIOGRAPHY

- [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.
- [Tectronix, 2013] Tectronix (2013). Power analysis of pwm motor drives. 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 inertial navigation. Technical report, Computer Laboratory, University of Cambridge.

Appendix A

Additional information

A.1 The Kalman filter

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

$$\hat{x}_k^- \quad (A.1)$$

$$\hat{x}_k \quad (A.2)$$

The *a priori* state in eq.(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), first of all the Kalman filter estimates a state by neglecting the process noise

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (A.3)$$

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

$$Q = E(w_k w_k^T) \quad (A.4)$$

$$R = E(v_k v_k^T) \quad (A.5)$$

APPENDIX A. ADDITIONAL INFORMATION

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

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

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

$$\tilde{y} = z_k - H\hat{x}_k^- \quad (\text{A.7})$$

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

$$S_k^- = HP_k^-H^T + R \quad (\text{A.8})$$

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

$$K_k = P_k^- H^T S_k^{-1} \quad (\text{A.9})$$

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

$$\hat{x}_k = \hat{x}_k^- + K_k \tilde{y}_k \quad (\text{A.10})$$

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.

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

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 rather intuitive. 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

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

Where

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

A.2. ARDUINO PWM IMPLEMENTATION

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

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

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

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

The innovation covariance mentioned in A.8

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

Kalman gain in eq. A.9

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

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

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

And then at last the error covariance

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

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 [Tectronix, 2013] and the human ear can not hear frequencies above ~20 KHz. The Arduino Uno's maximum PWM frequency is ~61 KHz, when set to high the motor

APPENDIX A. ADDITIONAL INFORMATION

controller consumes power so a good setting is just above hearing range. Which the motor control can handle [Pololu, 2014]. 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. Controls 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. 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 frequency on timer1 is 53 KHz), see AVR documentation for more info [Atmel, 2010].

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 loop and does not effect the main program but is rather run on a separate thread (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 was 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. Phase correct mode is useful when multiple PWM signal are used, but also was useful to achive the correct frequency in the particular case due to its counting mechanic.

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

$$frequency = \frac{clock}{2 \cdot prescaler \cdot TOP} \quad (A.21)$$

But the other pin (nr 3) is left unaffected. So if the desired PWM is 20 KHz it is done by choosing 100 as *TOP* (OCR2A for timer2) and the prescaler value

A.2. ARDUINO PWM IMPLEMENTATION

to 8 ($TCCR2B = (1 \ll CS21)$) pin 11 will always have 50% duty cycle at 20 KHz. The duty cycle for pin 3 can be set continuously. If another duty cycle for pin 3 ($OCR2B$) is set, the frequency will be maintained at 20 KHz due to the TOP ($OCR2A$) value is left unchanged. 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, the correct WGM values should be set in the $TCCR2A$ and $TCCR2B$ registers. The fifth 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 from the Lagrange equation [Block et al., 2007] acquire the equations of motion . Consider the Lagrangian equation

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

Where τ is the generalized 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.

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

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

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

$$\mathcal{L} = E_k - E_p \quad (\text{B.4})$$

$$E_k = \frac{I_c \cdot \dot{\theta}^2}{2} + \frac{I_f \cdot \dot{\phi}^2}{2} \quad (\text{B.5})$$

$$E_p = M_{tot} \cdot g \cdot l \cdot \cos \theta \quad (\text{B.6})$$

The Lagrangian (B.4) is then

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

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 at the center of the cube. Equation (B.2) and (B.3) with (B.4)

$$I_c \cdot \ddot{\theta} - M_c \cdot g \cdot l \cdot \sin \theta = -\tau_k \quad (\text{B.8})$$

$$I_s \cdot \ddot{\phi} = \tau_k \quad (\text{B.9})$$

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

$$I_c \cdot \ddot{\theta} - M_c \cdot g \cdot l \cdot \sin \theta = -I_s \cdot \ddot{\phi} \quad (\text{B.10})$$

To use linear control methods the model has to be linearised. This is done at the unstable 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

$$\sin \theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} \dots \approx \theta \quad (\text{B.11})$$

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

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

The transfer function that explains of ϕ as a function of θ is then

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

As the controller input is voltage, the angle of the cube has to be expressed with a voltage input. The torque exerted on the flywheel is described by the current flowing through the motor as well as the motor and gear efficiency

$$I_f \cdot \ddot{\phi} \cdot \eta_m \cdot \eta_g \cdot \Gamma = K_t \cdot i \quad (\text{B.14})$$

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

$$U = L \frac{di}{dt} + iR + \dot{\phi} K_e \quad (\text{B.15})$$

B.1. SYSTEM DYNAMICS

Equation B.15 and B.14 gives

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

Note that the motor inductance is 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.

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

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

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

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

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

TRITA MMK 2015:18 MDAB071