DEGREE PROJECT, IN MECHATRONICS, FIRST LEVEL
STOCKHOLM, SWEDEN 2016

Quadcopter

Sensor and filter evaluation

Axel Johansson, David Wallén

KTH ROYAL INSTITUTE OF TECHNOLOGY
ITM, SCHOOL OF INDUSTRIAL ENGINEERING AND MANAGEMENT

KTH Industrial Engineering and Management

Bachelor's Thesis MMKB 2016:18 MDAB079

Sensor and filter evaluation for quadcopters

Axel Johansson
David Wallén

Approved	Examiner	Supervisor
2016-06-07	Martin Edin Grimheden	Daniel Fede

ABSTRACT

In an electromechanical system hardware and software interoperate to perform a given task. Regardless of the task, it can be hard to know what level of complexity of components and software needed to perform well. Could you save on production costs by using cheaper sensors or is more expensive alternatives from well-known manufacturers needed? Does the control become simpler if you have more expensive sensors or is there always a need for advanced algorithms and filters?

In this report a quadcopter controlled by two different sensors and two different filters has been investigated. The two filters that were compered consisted of a simple complementary filter and a more advanced Kalman filter. The quadcopter was used as an empirical test to see how varying amount of disturbances or less exact data from the sensors affect a real system.

The results showed that the performance of the different sensors were quite similar, both in the area of raw data, calculated standard deviances and what was visually observable. With the constructed control system, the quadcopter was stabilizing around one axis with an accuracy of $\pm 4°$ and the cheaper sensor actually recorded a lower standard deviance of the angle (1.624° compared to 1.754° for the more expensive sensor).

This page has been intentionally left blank.

KTH Industriell teknik och management

Kandidatarbete MMKB 2016:18 MDAB079

Sensor- och filterundersökning för quadcopters

Axel Johansson
David Wallén

Godkänt	Examinator	Handledare
2016-05-09	Martin Edin Grimheden	Daniel Frede

SAMMANFATTNING

I ett system samspelar hårdvara och mjukvara för att tillsammans utföra önskad uppgift, men det kan vara svårt att veta vilken nivå på komponenter samt kod som krävs. Kan man spara in på produktionskostnader genom att använda billiga sensorer eller bör man investera i dyrare varianter från välkända tillverkare? Blir regleringen enklare om man investerar i en bra sensor eller behövs det alltid avancerade algoritmer och filter?

I denna rapport presenteras resultaten från undersökningar av en quadcopter som reglerats med hjälp av två olika sensorer samt använt sig av två olika filter för att filtrera indatat. De två filtren som jämfördes var ett enkelt komplementärfilter och ett mer komplicerat Kalmanfilter. Quadcoptern användes som ett praktiskt test för att se hur mycket störningar eller mindre exakt data från sensorer påverkar ett verkligt system.

Resultatet visade att prestandan för sensorerna blev väldigt snarlika, både gällande mätvärden, beräknade standardavvikelser och vad som var visuellt observerbart. Med den konstruerade regulatorn svängde quadcoptern (runt en axel) vid användandet av båda sensorerna mellan $\pm 4°$ fast den billigare sensorn hade en något lägre standardavvikelse av vinkeln (1.624° jämfört med 1.754° för den dyrare).

This page has been intentionally left blank.

PREFACE

The writers would like to thank Daniel Frede for excellent counseling and advises during the project and Martin Edin Grimheden for the opportunity to do this project.

Författarna vill tacka Daniel Frede för utmärkt handledning och generella råd på vägen samt Martin Edin Grimheden för möjligheten att genomföra denna projekt.

Axel Johansson, David Wallén

KTH, 2016-05

This page has been intentionally left blank.

CONTENTS

NOMENCLATURE

Explanations for symbols and abbreviations in the order they appear in the report.

Symbols

Symbol	Description
θ	Angle [deg]
$\dot{\theta}$	Angular speed [deg/s]

Abbreviations

Abbreviation	Desctiption
IMU	Inertial measurement unit
PID	Proportional-integral-derivative
DC	Direct current
PCB	Printed circuit board
I^2C	Inter-integrated circuit
MCU	Microcontroller unit
ESC	Electronic speed controller
PSU	Power supply unit
PPM	Pulse-position modulation
IDE	Integrated development environment
CPU	Central processing unit

1 INTRODUCTION

This chapter addresses the projects background, purpose and scope as well as the method used to carry out the project.

1.1 Background

The future is drones. Automated delivery directly to your home and the ability to travel over rough terrain and obstacles effortlessly using drones will open up a more fast-paced future. An early example of this is the Amazon Air Prime project where drones are used to rapidly deliver small packages. (Amazon Prime Air, 2016)

A quadcopter is a type of drone with four motors, each with a single propeller all mounted at the same distance from the systems center of gravity. By controlling the speed of the motors, and thereby their thrust, the quadcopter is able to fly in all directions at any given moment.

Advanced sensors and software combined with high performance hardware is critical when it comes to flying autonomous systems. By understanding how the programming and communication between the on-board computer and the motors is implemented the hardware can be utilized as efficiently as possible. It is important to understand and know what makes the difference in a system (both regarding hardware and software) to produce the most cost efficient drone without compromising its functionality. Since the price of the electrical components can vary a lot, significant savings can be made by selecting the appropriate components. This is especially important when mass producing since even a tiny increase in margin can result in great profits.

1.2 Purpose

The purpose of this report was to analyze how the quality of the inertia measurement unit, shortened IMU, and the complexity of the filter for the sensor signal affected a proportional-integral-derivative, shortened PID, controlled quadcopter. Different price-ranged IMUs was compared together and evaluated in terms of accuracy and performance. Different software filters consisting of a more complex Kalman filter and a less complex complementary filter were compared and measurements were taken to see how the stability was affected.

The study can be described by the following research question:

- How much does the price of the IMU and the complexity of the filter affect the performance of a PID controlled quadcopter?

The impact of vibrations from the motors on the system was also investigated.

1.3 Scope

To answer the research question and speed up the development process some open source libraries used for communication between the Arduino Uno and the IMU was used and will only be covered briefly. The scope also included building the quadcopter system and a testing rig for the stability measurements needed to evaluate the research question.

Considering the time frame and budget the investigation had to be limited. Two different filters and two different IMUs were selected and each combination of the two was tested several times, forming a matrix of four different configurations.

The two different IMUs were selected to be as different from each other as possible, different manufacturer and different components, whilst still keeping the same communication protocol with the main central processor unit, shortened CPU. The two filters were selected to be very different in complexity whilst still both being popular alternatives for quadcopters.

1.4 Method

After the test cases were designed and electrical components selected the intended quadcopter was first designed using 3D modeling software. A prototype of the quadcopter was then manufactured together with a testing rig, see Figure 1. The main purpose of the testing rig was to provide a stable and reproducible testing environment and also to increase safety for both the quadcopter and its operators.

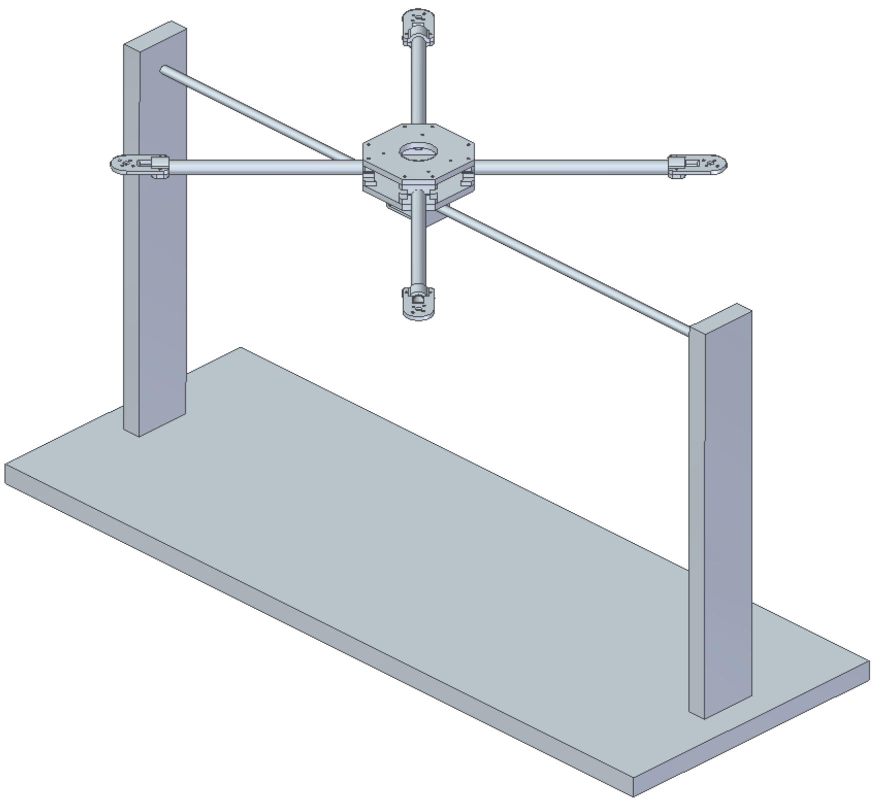

Figure 1. Test rig with quadcopter

Once the testing rig and the demonstrator were built the feedback controller was applied, tested and tuned, together with a customized software filter.

The comparison of the filters and sensors was done with the quadcopter fixed to the testing rig, only allowed to rotate around one axis. The filters were investigated by plotting the values read from the sensor over time and comparing them to both raw and filtered values whilst moving the drone in a standardized pattern. This was done both with the motors turned off and turned on. A good filter and IMU was indicated by a smooth output whilst a lesser filter had spiky and irregular output. The standardized pattern consisted of moving the quadcopter to one side until it hit a stopper in the rig (at $+90°$) and then to the other side (at $-90°$) and finally back to the initial position (at $0°$) within a timeframe of 15 seconds.

The final comparison was done by letting the quadcopter try to stabilize around one axis, whilst still being fixed in the testing rig. The quadcopter ran for one minute whilst the angle was measured and saved. Subsequently, the angle values were evaluated using Matlab (Matlab, 2016) and plotted with respect to time, which gave better overview of the stability. The standard deviation of the angle was also calculated for each run which gave an objective numerical indication on how stable the run had been. Several runs were made with each configuration, and the mean values were compared to give a fair and statistically reliable result.

All possible combinations of different filters and IMUs were tested and the results are presented in section 3.5.

2 THEORY

This chapter describes the different theories and components needed to answer the projects research question.

2.1 PID

A PID controller is a control loop feedback mechanism which calculates a value based on the difference between the desired value and the measured value. The proportional part (P) takes the current error value to calculate the output. The integral part (I) looks at past error values to calculate an output signal. The derivative part (D) uses the current change of the error value to calculate the output.

By calibration the three coefficients K_p, K_i, K_d the controller can make adjustments fitting to its purpose. A fast system without any overshoot is often impossible to create so a compromise regarding speed and stability has to be made. (Glad & Ljung, 2014)

The schematic of the control system can be seen in Figure 2.

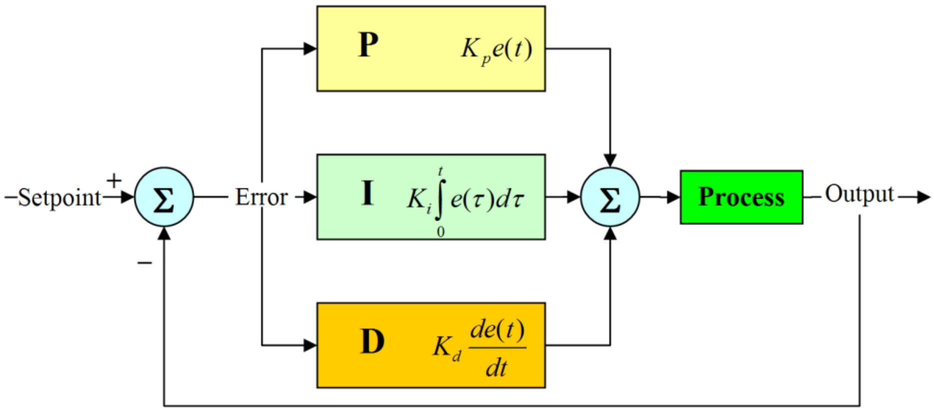

Figure 2. Flowchart for PID-controller (Nicisdigital, 2011)

2.2 IMU

The IMU is an electronic device that converts various physical conditions to electric signals which can be evaluated by the main processor. For the quadcopter the most important values are angle and angle rotation, both covering all three axes. This led to selection of 6-axial IMU measuring acceleration with and accelerometer and the rotation speed using a gyroscope.

The accelerometer contains a range of small capacitive plates, some fixed and some attached to springs. When acceleration forces (which are proportional to the acceleration) act on the plates they move and the change in distance between the plates result in a change of the capacitance, which is measured and communicated back to the main (Sparkfun, 2016).

The gyroscope works similarly to the accelerometer, but instead of plates it contains an H-shaped arm which vibrates in a certain direction (Epson, 2016). When the sensor is moved the direction of the vibration changes due to the Coriolis force, which is proportional to the angular speed (Wikipedia: Coriolis Force, 2016). The direction change is measured and communicated back to the main controller.

The communication for both the accelerometer and gyroscope can be either pulse-width modulated (as square waves with a varying duty cycle), analog (as a percentage of the supply voltage) or digital (using for example Serial Peripheral Interface or Inter-Integrated Circuit protocol), where the latter tends to be the most advanced and versatile option.

2.3 Filters

Filters are used to improve the accuracy of readings from sensors or other information sources. Vibrations and electromagnetic noise combined with incorrect readings from a sensor can make its data deviate from reality and even unusable, but with a filter a lot of these problems can be solved or reduced.

2.3.1 Rational filter

A customized filter was written in an attempt of investigating what was necessary for a filter in the quadcopter application and also simplifying the control process in general. A less complicated filter meant faster computation time and in return the sample frequency could be increased, to allow for greater precision.

The filter was dubbed "rational filter" as it takes physical limitations into account when evaluating output received from the IMU and discards irrational values likely caused by vibrations.

A maximum rotation that could be encountered by the quadcopter in flight was reasoned to be one full rotation per second. The maximum change per reading could be calculated by equation 2 and 3 and knowing the sample time a rational max could be set.

$$\Delta\theta_{max} = \frac{360}{\tau} \tag{1}$$

$$\Delta\alpha_{max} = \frac{360}{\tau} \tag{2}$$

The reasoning of the rational filter was that if the difference between a previous value and the newest value was larger than the allowed difference the new value was simply set to the old value plus (or minus) the maximum allowed difference. The filter can be expressed by the equation systems 3 and 4.

$$\begin{cases} \theta_{n+1} = \theta_n + \Delta\theta_{max} \ , \ \theta_{n+1} - \theta_n > \Delta\theta_{max} \\ \theta_{n+1} = \theta_n - \Delta\theta_{max} \ , \ \theta_{n+1} - \theta_n < -\Delta\theta_{max} \\ \theta_{n+1} = \theta_n \ , -\Delta\theta_{max} < \theta_{n+1} - \theta_n < \Delta\theta_{max} \end{cases} \tag{3}$$

$$\begin{cases} \alpha_{n+1} = \alpha_n + \alpha_{max}, & \alpha_{n+1} - \alpha_n > \Delta\alpha_{max} \\ \alpha_{n+1} = \alpha_n - \alpha_{max}, & \alpha_{n+1} - \alpha_n < \Delta\alpha_{max} \\ \alpha_{n+1} = \alpha_n, -\Delta\alpha_{max} < \alpha_{n+1} - \alpha_n < \Delta\alpha_{max} \end{cases} \tag{4}$$

where θ_n is the angular velocity and α_n is the angle read from the IMU.

2.3.2 Kalman filter

A more complex filter that was tested was the Kalman filter, which is an algorithm used to calculate a more precise state value by combining values from sensors and a predicted theoretical value based on the system's dynamic model. In the first step it predict measurements for cycle *k* with measurements taken at previous cycle, *k-1*. (Welch & Bishop, 2006)

The position and rotational velocity at the time *k* is represented by

$$x_k = \begin{bmatrix} \theta_k \\ \dot{\theta}_k \end{bmatrix}. \tag{5}$$

The predicted state for the system is calculated with

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \tag{6}$$

and the measurement model is represented by

$$z_k = Hx_k + v_k. \tag{7}$$

A and *B* are the matrixes;

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \tag{8}$$

$$B = \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} \tag{9}$$

and matrix *H* relates the state to the measurement z_k.

The Kalman filter uses a feedback control to predict the process state with it then compares to the measurements taken by the sensors. The equations used during the prediction phase are

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{10}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{11}$$

and the equations for correcting the estimation are

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{12}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{13}$$

$$P_k = (I - K_k H)P_k^- \tag{14}$$

Q and R are trust factors relating to the presumed accuracy of the measured values, called process noise covariance and measurement noise covariance.

2.3.3 Complementary filter

The third filter that was used was the simpler complementary filter. It combines the values from the gyro and the values from the accelerometer to get a more accurate estimation of the angle of the system as described in equation 15.

$$\theta_n = a \cdot \theta_{n,\ acc} + (1 - a) \cdot \left(\theta_{n-1} + \dot{\theta}_{n,\ gyro} \cdot dt\right) \tag{15}$$

Where a is a trust factor determined by how well the gyroscope and accelerometer performs (typically set to a <0.1) and θ is the angle of the system. This method is simple yet eliminates potential drift coming from the gyroscope and noise picked up by the accelerometer.

2.4 Brushless motors

The motors used on the quadcopter were brushless direct current, shortened DC, motors which mainly consist of two parts; the stator and the rotor, as seen in figure 3. The stator receives power which is run through coils to create electromagnets. There are fixed permanent magnets fixed on the rotor and by changing the current and thereby the electromagnets polarity torque can be created. A sensor is needed to match the current changes to the revolution speed. Alternatively, the back electromotive force in the undriven coils can be measured in order to determine the rotor position. (Radiocontrol wiki, 2016)

Using a brushless motor has many advantages. It has good weight to power ratio which is essential when it comes to flight and it runs both reliably and quiet. This is achieved by only needing to provide power to nonmoving parts of the motor and therefore less parts are needed.

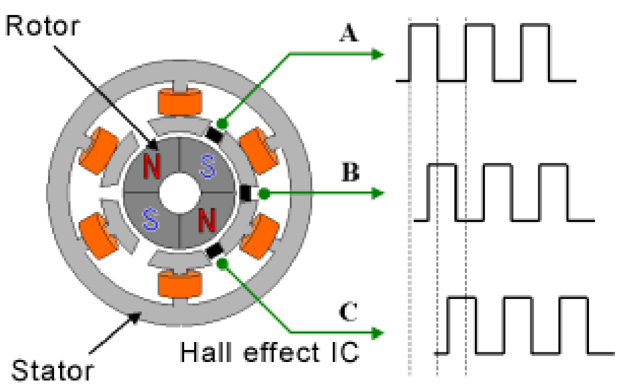

Figure 3. Basic structure of a brushless motor.

3 DEMONSTRATOR

This chapter addresses both the process of making the demonstrator as well as the result.

3.1 Problem formulation

Construction of the demonstrator involved a couple of different challenges, both mechanical and electrical, as listed below:

- The main body had to be durable and rigid while keeping a low weight.
- The arms extending to the motors needed to have a low surface area with minimal flex.
- Center of mass had to be in the center of the x- and y-axis to ease the maneuverability.
- The IMU had to be as centered as possible to reduce errors, especially when measuring acceleration.
- Motors and propellers had to be balanced to reduce vibrations and subsequent errors in measurement.

3.2 Electronics

All of the electronics were connected on a custom made printed circuit board, shortened PCB, which was positioned in the center of the drone body on top of the main control circuit, which was an Arduino Uno. Figure 4 shows the wiring diagram for the construction.

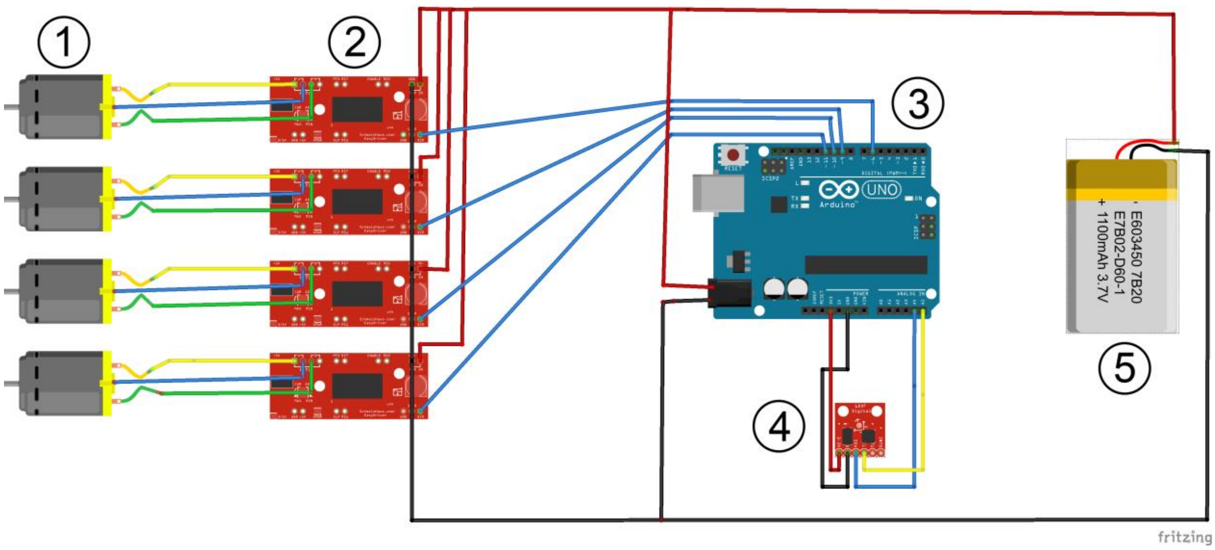

Figure 4. Drone control wiring diagram. 1: Motors, 2: Electronic speed control, 3: Arduino, 4: IMU, 5: PSU.

When the drone was in testing it was powered off a stationary power supply unit, shortened PSU, and the Arduino was connected to a computer for programming, debugging and collection of data.

3.2.1 Controller

An Arduino Uno was selected as the micro control unit, shortened MCU, for the control system due to its simple and quick integrated development environment (shortened IDE), high component compatibility and many open-source libraries. It is a prototyping platform based on Atmel's ATMega32 with both analog and digital in- and outputs. It has a separate chip for reprogramming the ATMega and communication with the computer is done serially via USB.

3.2.2 Sensor

Since the sensors were essential to the stabilization functionality they were selected with care. A 6-axis sensor consisting of a 3-axis accelerometer and 3-axis gyroscope was chosen since a combination between angle and angle velocity gives the opportunity to combine them and get a more stable approximation of the angle of the system than using only one type of sensor. A 9-axis sensor was also considered but due to budget and time limitations 6-axis sensors were considered enough.

The more expensive alternative was a MPU6050 from Olimex with a price tag of $20 whilst the cheaper option was a GY-85 which cost only $7.

Both of the IMUs communicated to the Arduino using I^2C which was an advantage since it only took two wires for the data transfer, although the IMU had be positioned close to the Arduino in order to minimize capacitance which could corrupt the signal. Another advantage with having the same communications protocol meant that switching between the different sensors was easier and it also gave a more accurate comparison since reading data suffered similar latency on both IMUs.

3.2.3 Motors

Four brushless direct current, shortened DC, motors were used to drive the propellers. Since the power source on the drone was a stationary PSU four Electronic Speed Controllers, shortened ESC, were used to convert the power from the PSU to the 3 channels of timed DC power that the motors needed. Each ESC was also connected to a digital output on the Arduino where a PPM signal was used to regulate the current to the motors. A kit consisting of four LDPOWER MT2206 motors (Hobbyking, 2016) and ESCs was acquired which contained the motors combined would give approximately 1,2 kg of thrust, well above the calculated weight of 0,6 kg for the entire quadcopter.

3.3 Software

The software was mostly written in Arduino's own IDE which is a C based platform that automatically compiles the code and programs the MCU.

In the analysis of different sensors and filter, the program Processing (Processing, 2016) was also used to graph values received from the IMU via the Arduino over serial connection in real time.

The main functionality of the program is visualized in Figure 5 as a flow chart.

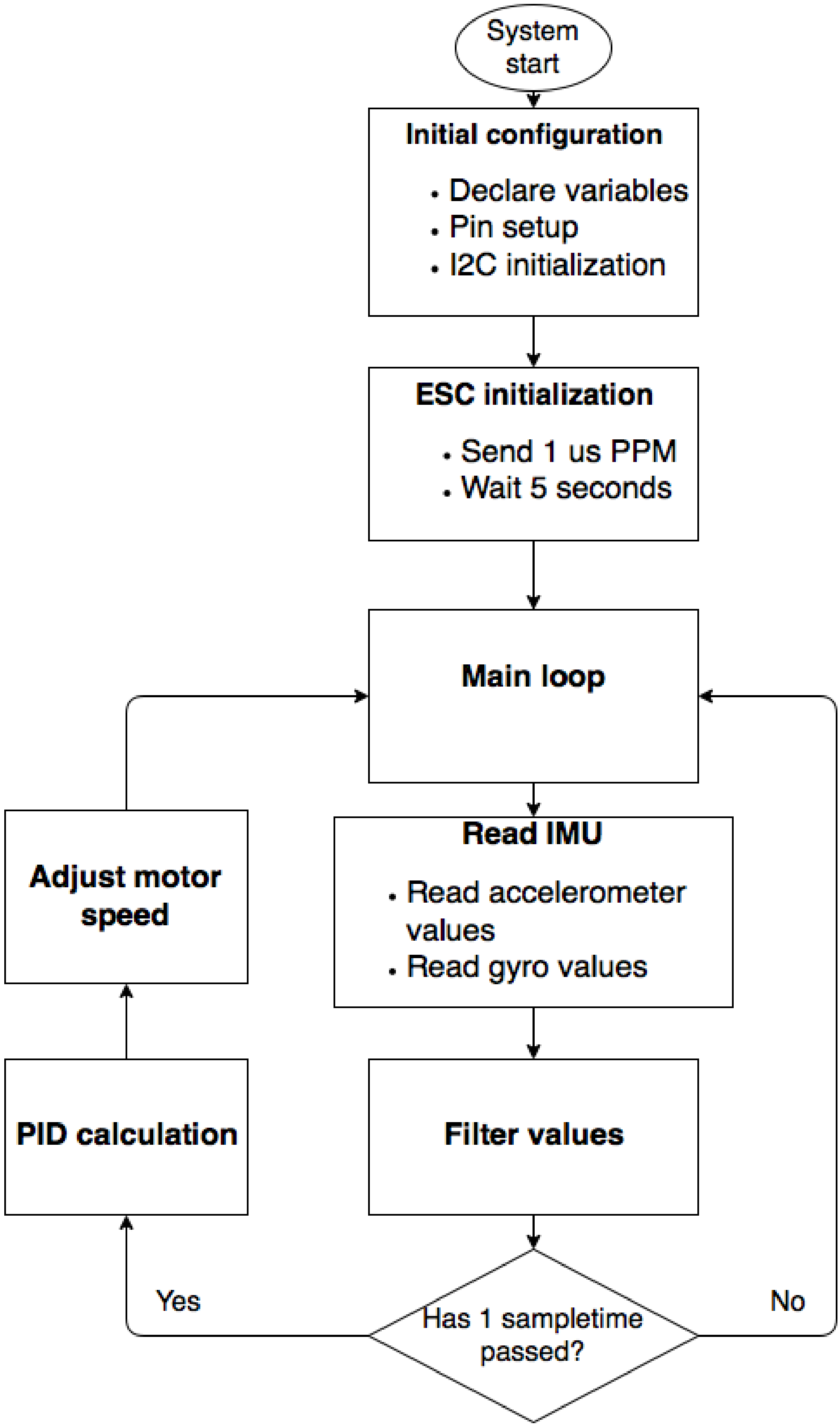

Figure 5. Program flow chart

3.4 Hardware

The drone was designed based on both studies of already existing models and personal preference. One of the first parameters that was fixed was the amount of motors used to generate lift. To simplify the control system and cut down on cost the number of motors was set to four and consecutively the design of the main body could begin. The first design of the drone was a completely symmetric 2D profile (to allow for easier control) which was laser cut out of wood and the motors were directly bolted onto the profile, as seen in figure 6. It was a simple design inspired by popular commercial drones as the "*Quantum Outlaw"* (Hobbyking, 2016). Wood was chosen at the early stages since it is fairly easy to work with whilst being both strong and light.

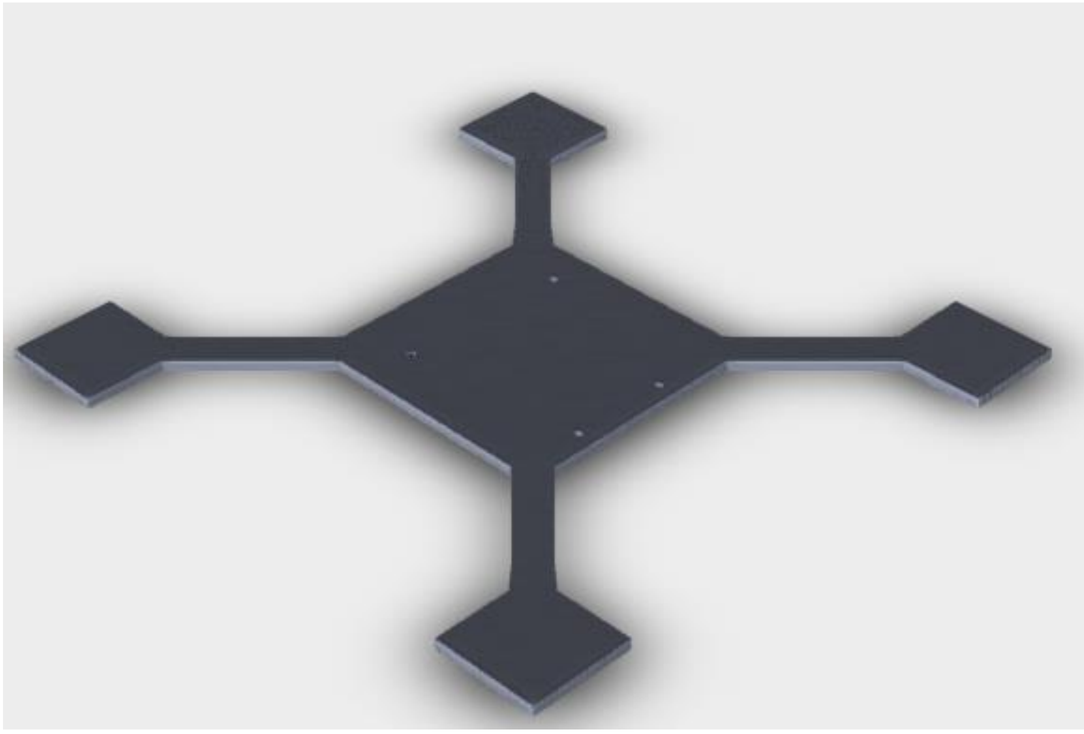

Figure 6. Profile of the first version of the drone.

In addition to the drone prototype a testing rig was also constructed, as seen in figure 1, to allow testing in various degrees of freedom and ease the development process. It also provided increased safety during configuration of the control system as the drone was prevented from lifting.

For the final design and even lighter and more rigid structure was desired. Four carbon fiber tubes extended from the main body, which was laser cut poly methyl methacrylate, also known as acrylic glass. The carbon fiber was even lighter than wood and the round profiles provided a much more rigid constriction. The motors were mounted on custom 3D printed plates on the end of each carbon fiber tube, as can be seen in Figure 7.

Figure 7. The final design of the quadcopter.

To reduce vibrations from the motors (and subsequent inaccurate readings from the IMU) the Arduino and the IMU were suspended using rubber bands such that there was no contact with the main body.

3.5 Results

Figures 8 to 11 display the testing related to vibrations in the structure using both hardware and software counter measures all run using the MPU6050.

As seen in figure 8 the accelerometer produced a less stable output and the gyro was more stable but instead drifted over time, which was expected from theory.

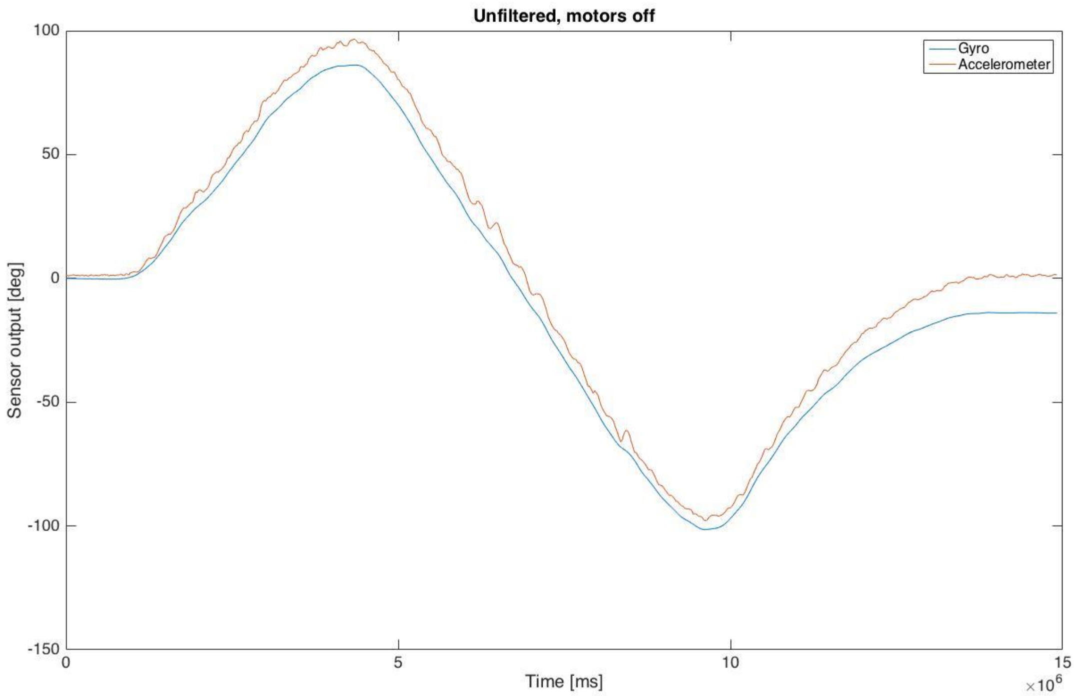

Figure 8. Gyro and Accelerometer values, motors off.

Figure 9 displays the motors turned on to a throttle level of 25 % and it is clear that the gyro value is unaffected by the vibrations, while the accelerometer is suffering from major disturbance.

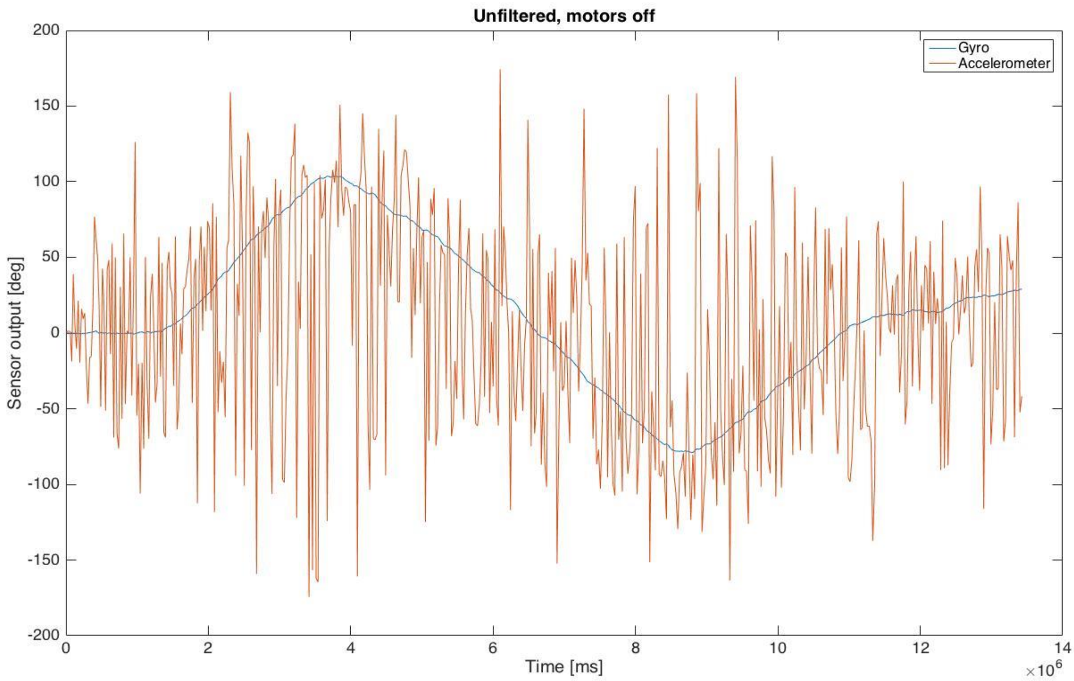

Figure 9. Gyro and Accelerometer values, motors on.

With both the rational and complementary applied the noise was significantly reduced, which can be seen in figure 10.

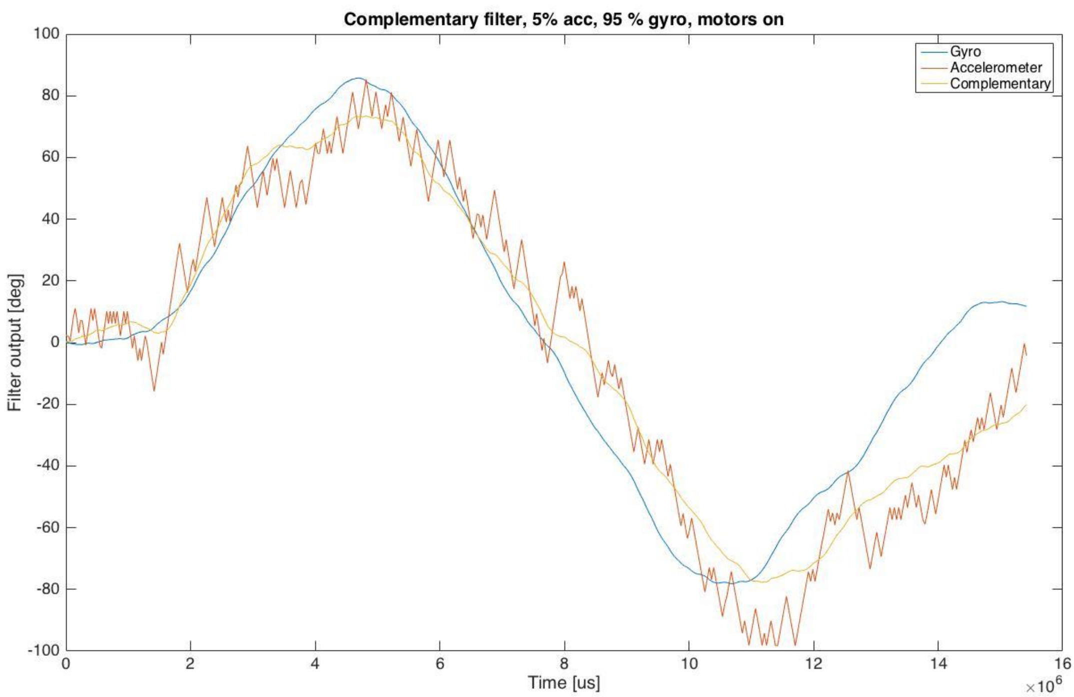

Figure 10. Rational and complementary filter, motors on.

However, with simple vibration dampeners the noise could be effectively reduced even without a software filter as seen in figure 11.

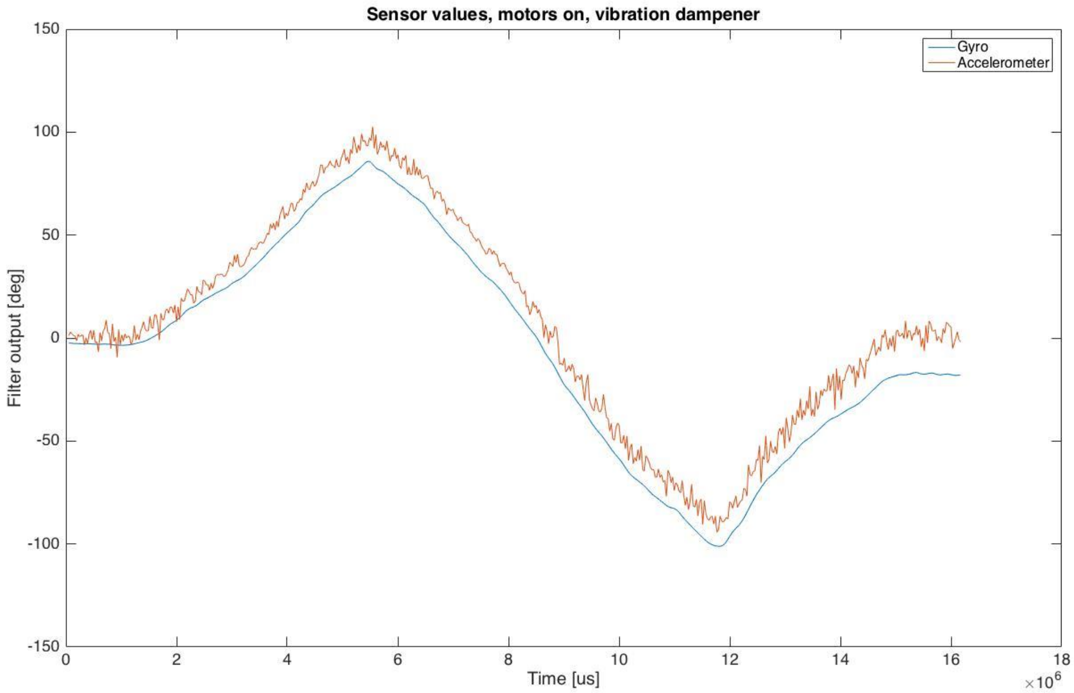

Figure 11. No filter, vibration dampener, motors on.

Figure 12 displays a comparison between a complementary- and Kalman filter.

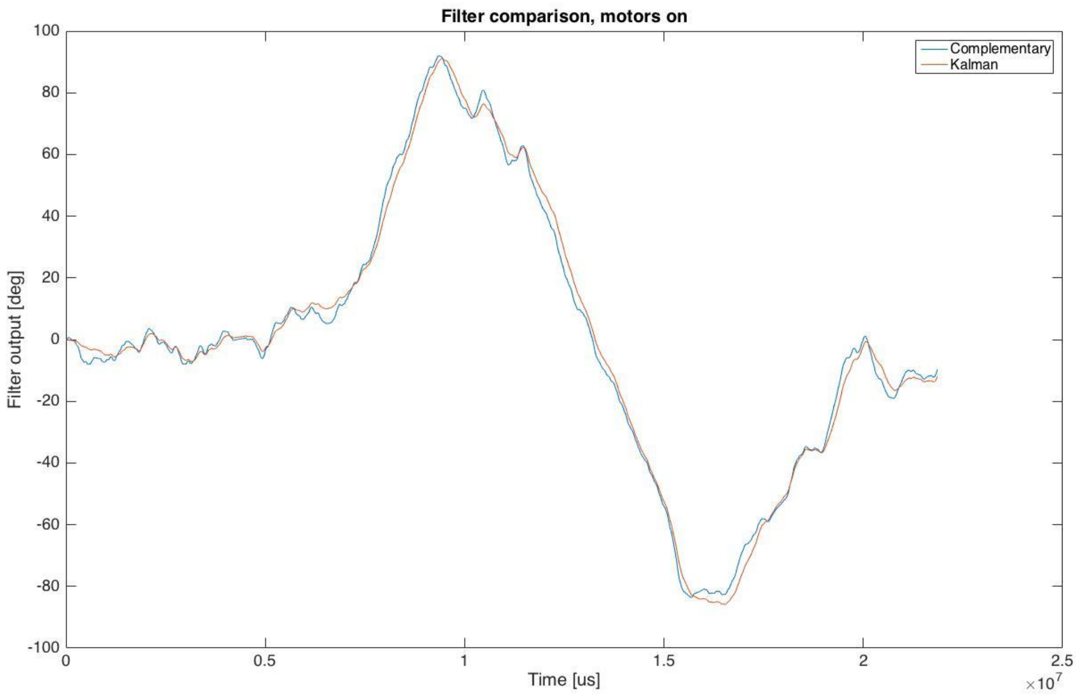

Figure 12. Complementary and Kalman filters.

Figure 13 displays the result of complementary filter combined with vibration dampeners.

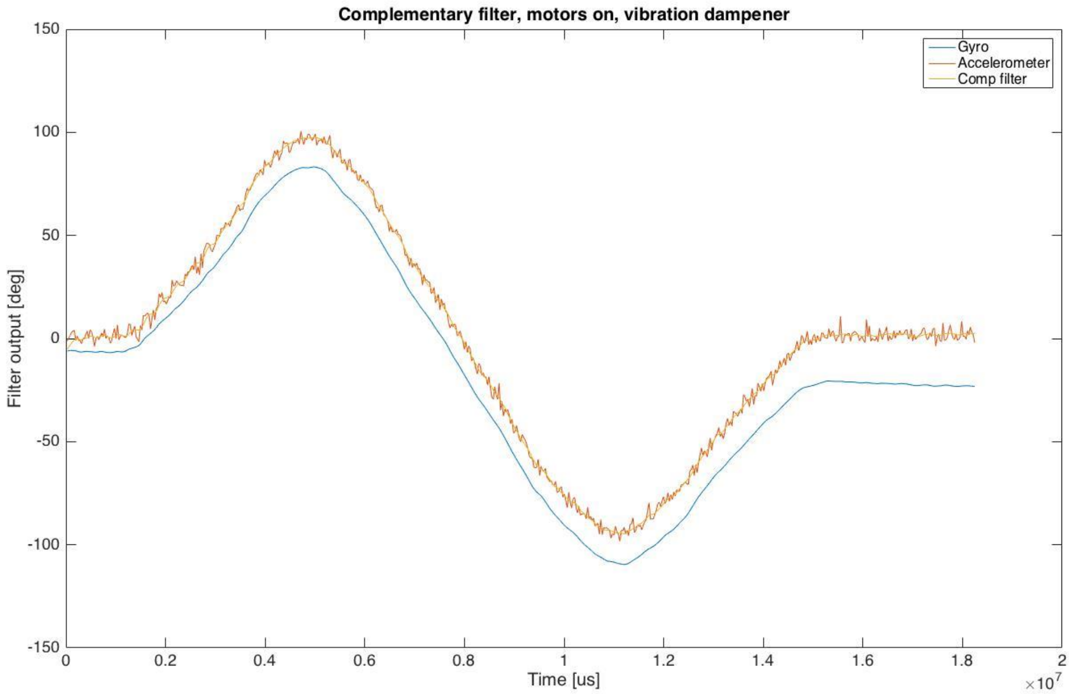

Figure 13. Complementary filter, motors on, vibration dampener.

The stability using the best possible conditions (filters and vibration dampeners) can be seen in figure 14.

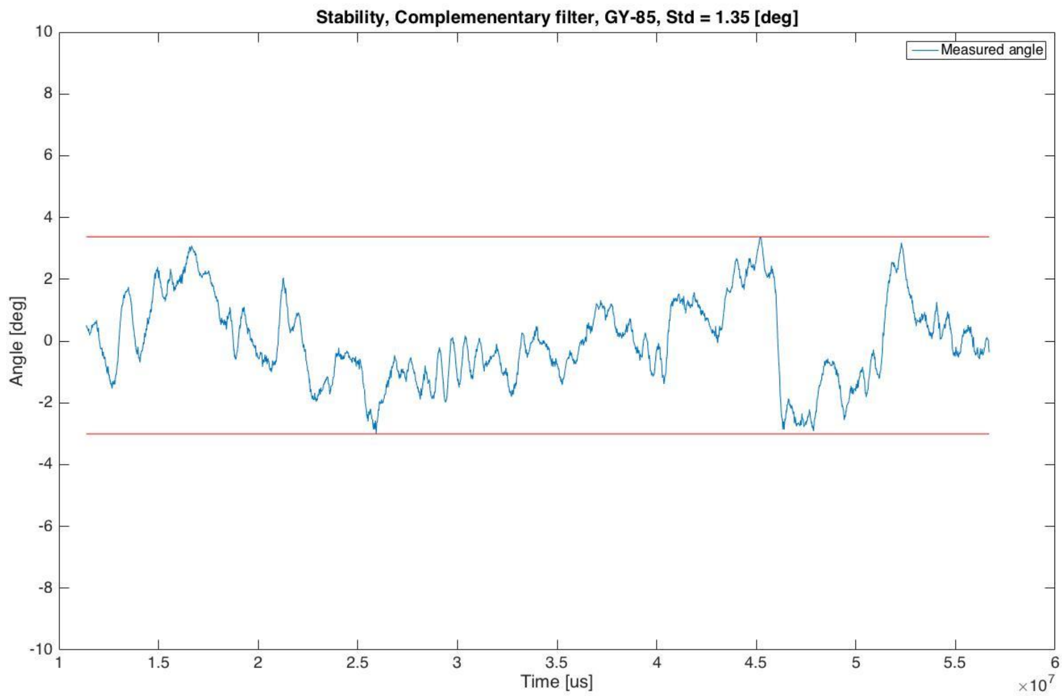

Figure 14. Stability, complementary filter, GY-85.

In total five runs were made with each sensor and filter combination and a mean value was calculated presented in table 1. All data for each run can be found in Appendix A under Collected data.

Table 1. Mean Standard Deviation of angle.

Sensor	No filter	Complementary filter	Kalman filter
MPU6050	Not stabilizing	1,754	2,104
GY-85	Not stabilizing	1,624	2,456

4 DISCUSSION AND CONCLUSIONS

This chapter discusses and summarizes the result presented in previous chapter. The conclusion is based on the result and aims to answer the research question.

4.1 Discussion

During the project several unforeseen aspects that affected the stability were encountered. To tackle the issues both software and mechanical solutions were attempted with varying results. The results of different IMUs and filters did not deviate as much as anticipated and led to some interesting conclusions.

4.1.1 Vibrations

As understandable from the figures in section 3.5, there was no use trying to control the quadcopter based on the unfiltered sensor readings or the mean value, as it provided a too coarse estimation. Overall there was far too much noise and the controller would have been very inaccurate. Even when the motors were running on as little as 25 % throttle level there was a dramatic increase in noise on the output from the accelerometer caused by vibrations from the motors and slightly unbalanced propellers.

The rigid carbon fiber body also led to an increase in noise compared to the wood prototype, but since wood is not a feasible material with regard to crash resistance it was not an option. The gyroscope was not as affected by the vibrations which led to reasonably stable output even with motors running at higher RPM.

First the rational filter was applied which significantly reduced the noise, however the mechanical solution with rubber band suspension was far superior and therefore it was used in all subsequent test scenarios.

4.1.2 Calibrating the filters

Both the complementary filter and the Kalman filter needed to be tuned to optimize their performance. This was done by empirically testing the system and iterating towards the best configuration meaning that there might exist better settings for both filters. The tuning becomes especially relevant when applying the filters to signals with a lot of noise and misleading values and will therefore not affect our results significantly after the point when vibration dampeners were implemented. Since the Kalman filter theoretically has the capability of filtering out noise to a greater extent than the complementary filter it could have been possible (with much greater effort put into tuning) to fly the quadcopter without the rubber bands as vibration dampeners. But it was deemed more reasonable to try to reduce the vibrations as much as possible with mechanical solutions before relying too much on the software, since having to compensate for a lot of noise makes the filter slower when it comes to pick up the actual changes of the system.

4.1.3 IMU comparison

This study was done using only two IMUs; one GY-85 and one MPU6050. To verify the results given more sensors (and more units of each model) should be tested. A sensor's performance could vary from unit to unit and drawing a conclusion based on two units is not recommended. However with regard to the limited time and money Inquiring more sensors was not possible due to limited money.

4.2 Conclusions

The research question for the project stated in chapter 1 states:

- How much does the price of the IMU and the complexity of the filter affect the performance of a PID controlled quadcopter?

According to the results presented in chapter 3.5 in table 1 the type of IMU did not significantly affect the quadcopters stability in this case. The two units performed at the same level (within a range of ± 8 %) and the results were deemed too be too similar to draw any conclusion whether one is better than the other.

Once vibrations had been mechanically reduced the difference between filters also became more subtle than expected, although the complementary filter did perform better, as can be seen in table 1. With a good and relatively smooth reading from both the accelerometer and gyroscope, see figure 11, the complementary filter, which relies almost completely on the gyroscope values with only corrections from the accelerometer, gave the best stability. During conditions with a lot of noise the Kalman filter has the possibility to function better, but with already accurate input a lot of its complexity is redundant and only makes the system slower.

This result lead to the conclusion that for a system which is heavily affected by vibrations it is much more efficient to try and reduce the vibrations mechanically before relying too much on the software. However once proper vibration dampening measures have been taken a correctly tuned filter and controller will lead to a very stable system fully capable of flight, even with cheaper IMUs and less complex filters.

When applied in mass production of quadcopters it could even be considered a recommendation to select a less expensive IMU since there is a significant price difference and not at all significant differences (within the scope of this investigation) in output stability. Selecting a cheaper IMU will result in continued cost savings throughout all the production and increase the profit margin. However, as mentioned earlier, if a less expensive IMU is to be used a test regarding its failure rate, factory errors and lifespan should also be performed.

When implementing the filter and controller in mass production the recommendation is to invest in deeper understanding of a complex filter (such as a Kalman filter) combined with proper testing and tuning of the software. Since this is a one-time investment that will grant instantaneous quality improvement and increased user experience of the quadcopter which adds product value and will in the long run generate more sales it is easy to motivate more effort (and ultimately money) towards this area.

To summarize this report in one sentence; it is not what equipment you have that matters, it’s how you use it.

5 RECOMMENDATIONS AND FUTURE WORK

5.1 Recommendations

The most mathematically complicated part of the software is the Kalman filter and it is recommended to gain a deeper understanding of the concept in order to utilize its full potential. Even though final tuning of the Kalman filter would ultimately be done by empirical tests, much like what has been done within this investigation, a deeper understanding could further improve the final result of the stability.

To decrease the time it takes to calibrate the PID-controller buttons could be implemented to provide the option of quickly being able to change the parameters while the quadcopter is running. This could also give the user a better understanding and feeling for the different parameters.

5.2 Future work

To further investigate the relationship between differently priced sensors a wider range of sensors should be tested. Possibly also several units of the same model in order to detect any deviations. Analyzing multiple units of the same model would also give a more general indication of the quality of the IMU as the researcher could discover that there is a difference in percentage of correctly working units, a factor which also is to be considered especially when using a component in mass production.

It is also recommended to construct an external system for measuring the angle of the quadcopter to avoid burdening the main CPU with the task of extracting measurement data, something it will not have to do during normal operation. This would also ensure a more accurate reading.

REFERENCES

Amazon Prime Air. (2016, 03 30). Retrieved 04 11, 2016, from Amazon: http://www.amazon.com/b?node=8037720011

Epson. (2016, 04 02). Retrieved from www5.epsondevice.com: http://www5.epsondevice.com/en/information/technical_info/gyro/

Glad, T., & Ljung, L. (2014). *Reglerteknik.* Lund: Studentlitteratur AB, Lund.

Hobbyking. (2016, 03 14). Retrieved 05 03, 2016, from www.hobbyking.com: https://www.hobbyking.com/hobbyking/store/uh_viewItem.asp?idProduct=84722

Hobbyking. (2016, 02 28). Retrieved 05 03, 2016, from www.hobbyking.com: http://www.hobbyking.com/hobbyking/store/__85287__Quanum_Outlaw_270_Racer_Quadcopter_Frame_Kit.html

Matlab. (2016, 05 04). *Matlab*. Retrieved from Matlab: http://se.mathworks.com/products/matlab/

Nicisdigital. (2011, 06 27). Retrieved 04 26, 2016, from Mr Digital: https://nicisdigital.wordpress.com/2011/06/27/proportional-integral-derivative-pid-controller/

Processing. (2016, 03 15). Retrieved from Processing: https://processing.org/

Radiocontrol wiki. (2016, 04 10). Retrieved from Wikia: http://radiocontrol.wikia.com/wiki/Brushless_motor

Sparkfun. (2016, 04 12). Retrieved from www.sparkfun.com: https://learn.sparkfun.com/tutorials/accelerometer-basics

Welch, G., & Bishop, G. (2006, July 24). *Department of Computer Science.* Retrieved from www.cs.unc.edu: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

Wikipedia. (2016, 04 24). Retrieved 05 03, 2016, from www.wikipedia.org: https://en.wikipedia.org/wiki/Brushless_DC_electric_motor

Wikipedia: Coriolis Force. (2016, April 30). Retrieved from www.wikipedia.com: https://en.wikipedia.org/wiki/Coriolis_force

APPENDIX A: COLLEDTED DATA AND PARAMETERS

Collected data

Data from the five test runs of the quadcopter in the rig, standard deviation from 0°.

MPU6050	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Comp	1,62	1,93	1,81	1,62	1,79
Kalman	2,14	2,16	1,99	2,47	1,76

GY-85	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Comp	1,83	1,73	1,37	1,60	1,61
Kalman	2,31	2,48	2,65	2,46	2,38

Parameters

Parameters that were used in the project and their corresponding value.

Parameter	**Value**
K_p	0.7
K_d	0.3
K_i	0.09
a	0.01
τ	0.00001