Construction of a quadroter platform and an investigation of inertial navigation

CHRISTOFFER CEDERBERG
ANDRE ERICSSON
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY |
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Construction of a quadroter platform and an investigation of inertial navigation

Christoffer Cederberg and Andre Ericsson

Department of Electrical and Information Technology
Lund University

Advisor: Stefan Höst and Bertil Lindvall

February 23, 2017

Printed in Sweden
E-huset, Lund, 2017

Abstract

The usage and potential usage of unmanned aerial vehicles (UAV) in our society grows rapidly and therefore an interest of how to design such a craft was sparked. The ability of an autonomous flight is valuable in many applications and requires that the craft is equipped with a GPS-module. If the connection is lost the UAV's itinerary may differ from the predetermined one. Because of this an alternative navigation technique would be interesting. By measuring the inertia of an object it is possible to derive the location in relation to the starting point.

In this thesis report a comprehensive description of how a quadrotor helicopter is designed and constructed will be given. Furthermore, inertial navigation implemented by miniaturized mechanical and electro-mechanical elements will also be treated in this report.

Popular science article

As unmanned aerial vehicles (UAV) are widely used both in the private sector and in commercial businesses as tools for a variety of purposes, it would be interesting to take a closer look at both the construction and the ability to navigate such a craft autonomously without GPS. The vehicle which was built from scratch during this thesis is a helicopter with four motors / propellers, a so called a quadrotor helicopter, and the navigation technology is based on measuring the vehicles acceleration as a function of time, i.e., inertial navigation.

Quadrotor Helicopter

A quadrotor helicopter is, as most people know, a craft with four motors and propellers. With these it is possible to control the vehicle in any direction and the position about its vertical axis. These maneuvers are carried out by pairwise changing the rotation speed of the motors, which give rise to a force difference that will affect the angular position of the helicopter. If the craft is not parallel to the horizontal plane it will move in the same direction it leans towards.

The angles θ, ϕ and ψ, see *Fig.* 1 , are used to define the state of the helicopter. The signals that describes the angles were created by complementing the signals from three gyroscopes with two accelerometers and one, for the purpose designed, compass comprised of two hall effect sensors. The sensor algorithms were done with analog electronics.
As there are no external forces that bring the craft back to its initial position when it is subjected to an involuntary change in angle there is a need of control algorithms in order to gain stability. A control algorithm have a target value (the desired angular position for the helicopter) and a reference value (the measured angular position of the helicopter) as inputs and the output is the control signals for the motors, which corrects potential errors in the angular inclination. In this project the stability control was implemented by three cascade control structures, one for each axis, and were executed by a dual-core microcontroller on board the helicopter. The signals that describe the angles were sampled several hundred times per second and the control algorithms were executed with the same fre-

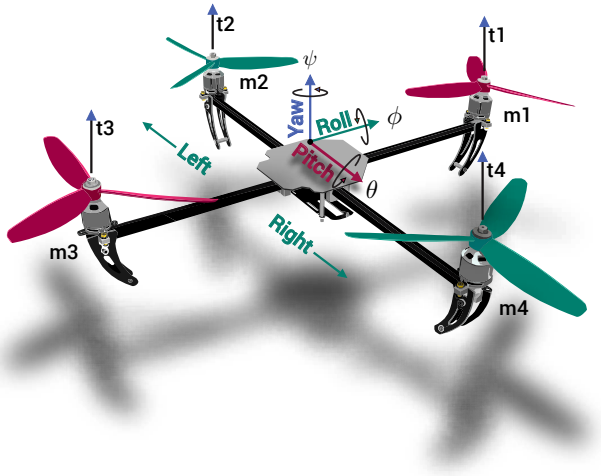

Fig 1: Quadrotor helicopter reference.

quency.

To change the angular inclination of helicopter the user adjusts the corresponding target values via a radio link. To send these values from a safe distance a remote controller is needed. The controller must have appropriate interfaces to set the desired angles θ, ϕ and ψ and the altitude of the craft. A remote controller for a game console meets these requirements. The casing and the joysticks of such a device was used while the remaining content was replaced with suitable components (processor unit, radio link, etc), see *Fig* 2.

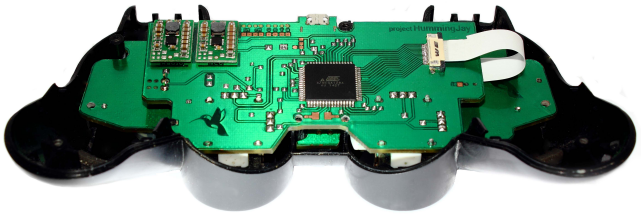

Fig 2: Modyfied remote controller .

Inertial navigation

The basic principle of inertial navigation is to measure the acceleration vector (acceleration and direction) that an object is exposed to, if this is known the exact position can be determined with a few mathematical operations. It is thus com-

pletely independent of external references. This technique has long been used by both military and in commercial applications. This has been done with the help of mechanical contraptions that are as expensive as advanced. Another problem is that they are often relatively large and therefore not suited to be carried on board a small remote-controlled helicopter.

Today the development of small and reliable sensors with low power consumption and high accuracy is done at breakneck speed and is used in variety of applications. It would therefore be interesting to investigate the sensors of the appropriate type with inertial navigation in mind. This requires a processor unit that can collect the measured values and perform the necessary calculations. Imperfections of the sensors and the processing unit's analog-to-digital converter of the processor unit yielded a flawed result of the positions. This error sometimes grew with time and thus limited the usability. On the other hand the technique showed good repeat accuracy of different lengths and accelerations, but for shorter time intervals.

An example of a test run is shown in *Fig* 3 where the calculated distance was 0.76 m and the actual distance was 0.75 m. Another calculation showed a normal flight of 100 m normally gave an error of less than 1 m.

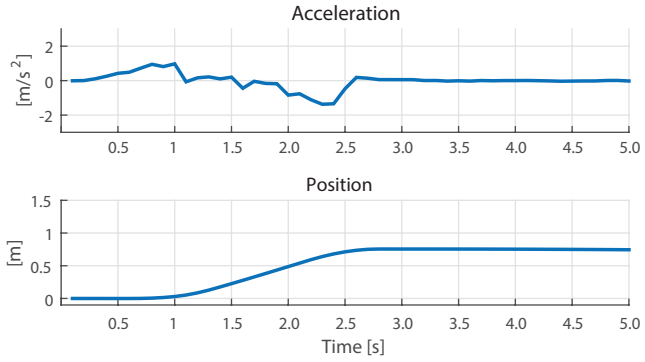

Fig 3: Examples of measured data and calculated position.

Acknowledgments

The authors would like to take the opportunity to express great gratitude towards Bertil Lindvall, Stefan Höst and Fredrik Cederberg for help and guidance. During this thesis the two students have obtained practical experience that is of tremendous value to them.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

We live in an age where drones are becoming abundant and have the ability to perform numerous tasks in a variety of fields such as, parcel delivery, surveillance, inspections and so on. In the majority of this applications it is required that the drones position is known and they are therefore equipped with devices that enables GPS-navigation. This type of positioning system has its limitations, such as latency and accuracy and poses one definitive demand that the craft is located outside of regions in radio shadow. If the crafts desired itinerary does enter such a region the knowledge of its position is lost. At this moment the question of what might replace or supplement such a system in a drone arises. A feasible solution is inertial navigation which is fully self contained and does not rely on any external references, it is only based on measuring the acceleration vector as a function of time. If this vector is known at all times it is possible to determine the location of the craft. This technique has long been used by the military but the construction is relatively large, heavy and expensive due to its mechanical complexity.

In today's market there are many and increasingly affordable MEMs (Microelectromechanical systems) sensors which makes it possible to measure the described vector. These sensors suffers from imperfections, that is for example, the zero reading drift, which is sufficiently large to reduce the usability when the sensor is incorporated in a navigation system. There are also enormous amounts of inexpensive microcontrollers, with good computational performance. By this fact another question arises, namely, to what extent is it possible to determine a drones position implemented by relatively simple means?

By designing the the drone platform from scratch, the authors will have the desirable freedom to manage and control all the necessary behavioural and functional properties of the craft. Desirable properties and behaviour to control would be

- weight
- flight characteristics
- driving system

- traction force
- flight time
- ease of replacing parts
- ease of improving parts and functions.

Thus the main goal of this thesis can be concluded to the following two parts. First the authors will create a drone platform, including a remote controller, that could be equipped with an inertial navigation system. The platform had to be created by relatively simple means and with a restricted budget. The second part will be to investigate to what extent a inertial navigation system can be used on a quadrotor platform. An initial goal of the desirable precision regarding the navigation was set to one percent, i.e. how far is it possible to move without exceeding the given tolerance.

1.1 Quadrator platform – Fundamental description

A quad rotor helicopter is an aerial vehicle that consists of four motors placed on a cross frame.

The angles θ and ϕ, which corresponds to the pitch and roll axis, are controlled by an individual motor pair, m_1 and m_3 for θ and m_2 and m_4 for ϕ, see *Fig* 1.1. When a difference between the generated thrust of the motors in a pair is present, the craft is subjected to an angular acceleration since the thrust is a force which is applied away from the center of gravity.

Every motor produces a reactive torque, q_1, q_2, q_3 and q_4, see *Fig* 1.2 below. To prevent the vehicle from drifting about its vertical yaw axis, which corresponds to the angle ψ, each motor pair is set to rotate in opposite directions.

Basic manoeuvring

The platform is in its equilibrium state, i.e hovering, while the pitch and roll axis are in parallel with the horizontal plane and the craft is not spinning about its vertical yaw axis. Under these circumstances θ and ϕ are equal to zero, ψ is constant and the sum of the thrust t1, t2, t3 and t4 is equal and opposite to the gravitational pull on the craft.

While θ or ϕ is not equal to zero the craft will perform a lateral movement. In

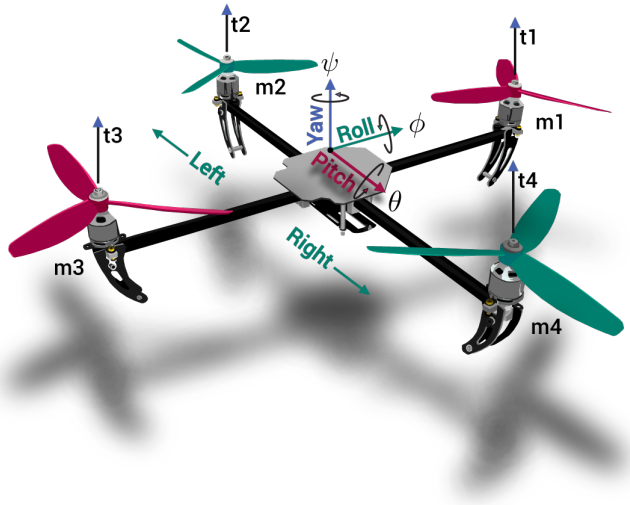

Fig 1.1: Quadrotor platform reference.

order for the craft to maintain the altitude while varying the θ or ϕ the sum of the generated vertical thrust component has to be constant. By increasing the angular velocity of one, and only one, motor pair the craft will spin around the yaw axis in the opposite direction of the blade travel, this due the reactive torque. See *Fig* 1.2. And again, the sum of the vertical thrust component contributed by each motor has to be constant while performing this manoeuvre if the vehicle has to remain at the same altitude.

Equations for the motor control signals

The control signal for the motors can now be deducted and are shown below.

$$y_{m_1} = x_{thrust} + x_{pitch} + x_{yaw} \tag{1.1}$$
$$y_{m_3} = x_{thrust} - x_{pitch} + x_{yaw} \tag{1.2}$$
$$y_{m_2} = x_{thrust} - x_{roll} - x_{yaw} \tag{1.3}$$
$$y_{m_4} = x_{thrust} + x_{roll} - x_{yaw} \tag{1.4}$$

Where x_{thrust} denotes the contribution from the operator and x_{pitch}, x_{roll} and x_{yaw} is proportional to the desired angles θ, ϕ and ψ.

In order to subject the platform to a lateral movement one of the following pair

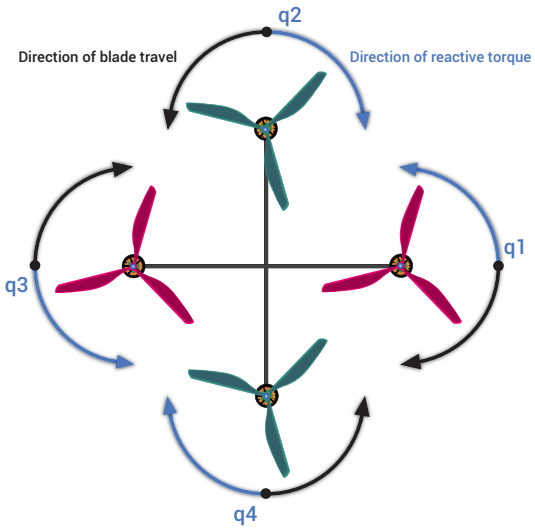

Fig 1.2: Top view of the quadrotor platform.

wise inequalities has to be true.

$$x_{pitch} < 0 \Rightarrow \text{a forward movement}$$

$$x_{pitch} > 0 \Rightarrow \text{a backwards movement}$$

$$x_{roll} < 0 \Rightarrow \text{a movement to the left}$$

$$x_{roll} > 0 \Rightarrow \text{a movement to the right}$$

Stability of the platform

A system is stable if it produces a finite output while it is subjected to a bounded input. This is the case as the poles of the systems closed loop transfer function resides in the left-half s-plane. A more practical approach of defining stability is to say that stability is a measure of the tendency of a system's response to return to zero after a perturbation.

By considering the following two different systems one can visualize the concept of stability. The first system is a damped harmonic oscillator, see *Fig* 1.3. If this system is subjected to a bounded disturbance it will in all cases return to its initial position. Hence it is stable.

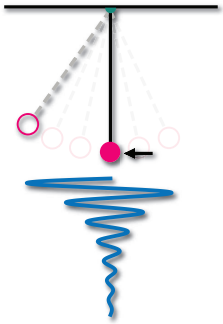

Fig 1.3: A damped harmonic oscillator.

The second system is an inverted pendulum. It is possible to find a position where the pendulum is at rest in an equilibrium. In this position even the smallest perturbation will make the pendulum fall and it will never return to its initial position, for a graphical reference see *Fig* 1.4. If the goal is to keep the pendulum in the equilibrium position one has to make the system stable with an active control. This can be realized with a control algorithm, sensors and actuators.

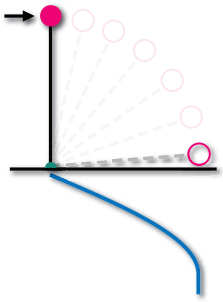

Fig 1.4: Inverted pendulum.

Now it is easier to understand that quadrotor platform is inherently unstable because of the fact that there are no external forces (except the gravity, and this force could yield a disastrous result) that will compel the vehicle to a stationary state while it is in the air. Imagine a situation where all four motors generate an equal amount of thrust and the sum is equal and opposite to the gravitational pull. The craft will of course be aloft but even the slightest perturbation in θ and/or ϕ will change its angular state and never again return to the initial angles. Thus one can conclude that this is a system in need of an active control.

Since the three axis, pitch, roll and yaw are orthogonal, each axis can be treated as a separate process with no correlation to each other. Therefore three one-variable controller schemes can be utilized to gain stability while the quadrotor platform is aloft.

1.2 Inertial Navigation

As previously described, an Inertial Navigation System (INS) is a system that navigates solely by measuring the inertia and it is fully self-contained. If one knows the acceleration vector and elapsed time since departure, the position is fully determined. The sensors utilized to carry out these measurements are accelerometers and gyroscopes.

Two different approaches

There are two main approaches to construct a navigation system of this kind, Stable Platform Systems and Strapdown Systems.

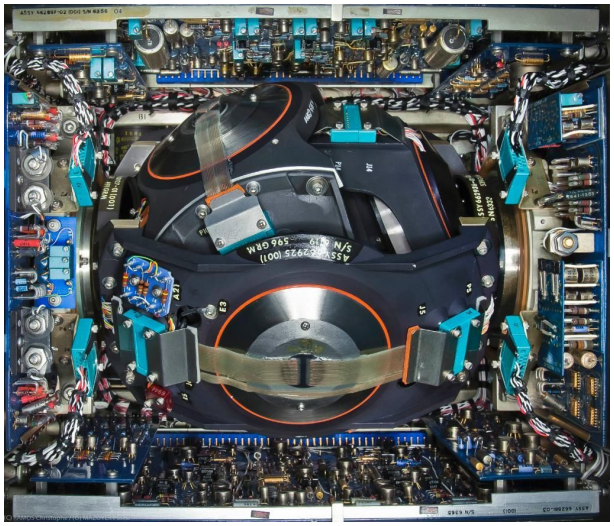

Fig 1.5: A stable platform.

In a stable platform system the sensor array of the moving craft, i.e the navigation systems frame is in parallel with the global reference frame at every point in time. This is done by using gyroscopic elements, i.e. utilizing the preserving effect of the torque of spinning objects. By measuring the moving crafts deviation from the stable platform the heading can be determined. See *Fig* 1.5.

In a strapdown system one has not the luxury of having a three-dimensional "compass" at your disposal. The navigation system's frame is not aligned with the global reference frame. This kind of navigation system has one advantage over the previous, which is that it is not mechanically advanced. The main disadvantage is that it is much more computational demanding.

1.3 Sensors

The sensors used to determine the angular inclination and position of the quadrotor platform will be based upon the MEMS technology. Below follows a brief description of this technology.

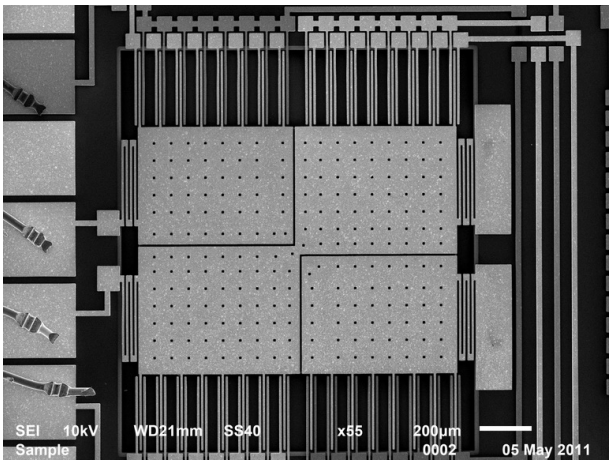

Fig 1.6: Differential MEMS accelerometer.

A MEMS sensor is comprised of miniaturized mechanical and electro-mechanical elements. The actual size of the micromachined elements varies but can be as small as one micron and below. The MEMs technology has be proven cost effective because it utilizes parts of the same fabrication process as integrated circuits which is well established in today's industry. The microstructures are mainly created by etching patterns in a silicon wafer. See *Fig* 1.6.

When it comes to performance, the micro-scale sensors has in plenty of cases surpassed the macro-scale counterparts whilst the typical per-device power consumption is smaller. Furthermore it has shown good reliability. These benefits make this sensor technology a much compelling choice, especially in mobile applications. This is why the authors have chosen to utilize and investigate this

sensor type. The sensors that will be used both for the stability control of the platform and the navigation system are gyroscopes and accelerometers. Furthermore it will be shown that a magnetometer is required. This sensor will not be of the MEMs technology.

Chapter 2

Quadrotor platform

In this chapter the required main components of the quadrotor platform, such as the remote controller, motors, sensors radio link, processing units and cross frame will be discussed and in some cases examined more closely. The implementation of the software for the platform controller and the quadrotor will be described. Furthermore the stability control algorithm and some necessary digital signal processing will also be addressed here.

2.1 Remote controller

The quadrotor platform needs to be operated, this will be done via a remote controller. The controller should be able to steer the craft in three dimensions. A controller intended for a gaming console would meet the criteria for the supposed purpose since they usually consists of two joysticks and several buttons. The two joysticks is ideal for controlling the thrust and the lateral movements, the buttons would be good to incorporate for extra functionality such as entering different modes (calibration etc).

The authors acquired a Playstation 3 (PS3) controller and emptied it, copied the dimensions of the original two circuit boards, one for the tactile buttons and one for the processing unit and joysticks. The joysticks used here are simply two potentiometers (per stick), one register the push/pull movement and one for the left/right. Besides of giving the quadrotor platform commands it would be beneficial to receive some information. As it turns out the PS3 controller has the capability of producing a haptic feedback via two DC motors. One of those was kept. The controller is also capable of giving a visual feedback via LEDs. This feature was also desirable and kept in mind. Further more the battery was also saved (3.7 V Li-Ion with a capacity of 1000 mAh).

Processing Unit

The tasks to be performed by the controller's processing unit is not comprehensive as it only involves converting the analog voltage from the joystick poten-

tiometers into the digital domain and polling the buttons and pass along the information to the radio transceiver via UART. Aside from this it also needs to control the motor and LEDs, preferably by a pulse width modulation (PWM) peripheral.

A 8-bit AVR from Atmel, the ATmega128, has all of the above mentioned features and was therefore chosen. See *Fig* 2.3 for a block overview.

Before the remote controller software is described it is convenient to explain the data transfer protocol.

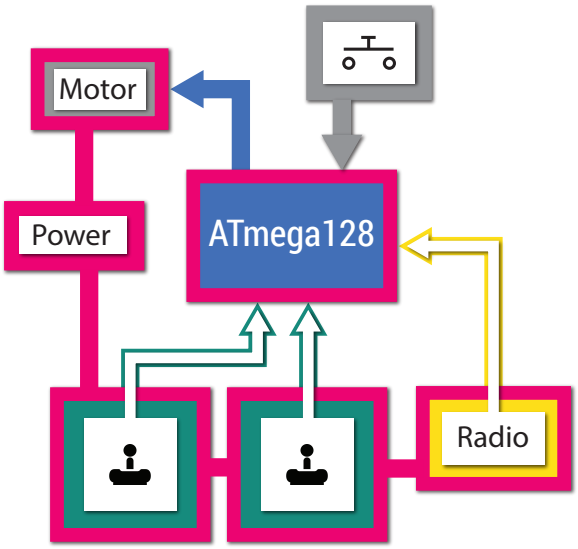

Fig 2.1: Block overview of the quadrotor platform controller.

UDQPP - Ultra Dense Quadrotor Platform Protocol

The protocol was designed with the intention to receive/send a complete command for the platform in one byte. The 3 most significant bits of the byte is the header and the 5 remaining bits are the payload. This allows for 8 unique packets and a payload value range from 0 to 31. The headers were assigned as showed:

000 = Thrust
001 = Counter-Clock-Wise Yaw
010 = Clock-Wise Yaw
011 = Forward Pitch
100 = Backward Pitch
101 = Left Roll
110 = Right Roll
111 = Buttons

By utilizing this protocol the platform operator has a resolution of 0.63° when the platform is limited to a maximum inclination of 20°. This translates to that any lateral acceleration smaller than 0.11 m/s^2 can not be compensated manually, this is worth to keep in mind. The equations used to approximate the acceleration will be showed in the following chapter.

Software

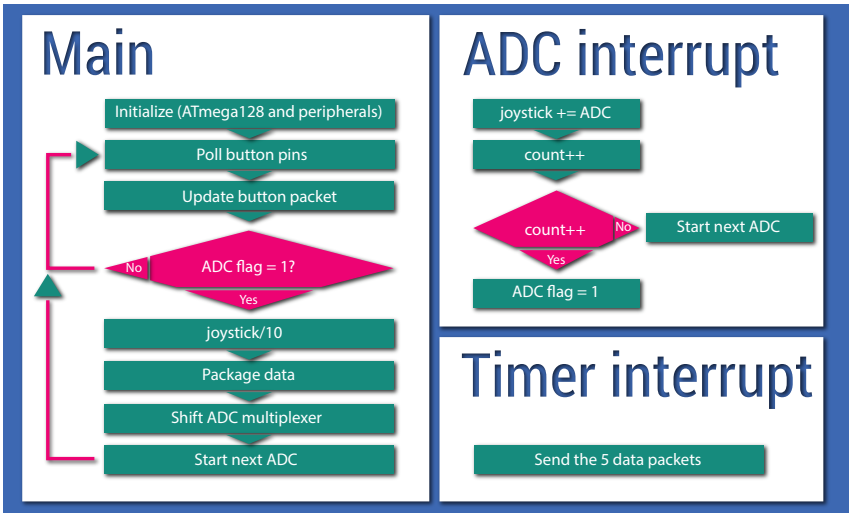

Fig 2.2: Flowchart for the quadrotor platform controller.

The processing unit performs its tasks in the following way. The ADC is initialized so that an interrupt service routine (ISR) is invoked after every completed conversion. The ISR accumulates the result (8-bits) in a variable, increment a counter and starts the next conversion. When 10 samples are done a flag is set

to 1 and no more conversions are done. The main loop polls the button pins and updates a variable with the button status. After this the previously mentioned flag is checked, if it is 1 the accumulated ADC value is divided by 10 and shifted 3 bits to the right so that the data fits in 5 bits and then put in the appropriate data packet. When this is done the ADC multiplexer is shifted and a new series of data conversions are carried out.

A timer is initialized to generate an interrupt at every timer overflow. Every time the timer ISR is triggered it sends 5 data packets to the radio transceiver. (It is only necessary to send 5 packets every time, one for each action, namely thrust, pitch, yaw, roll and buttons).

Please view the software flowchart in *Fig* 2.2 for a graphical reference.

Construction

The interior of the platform controller consist two PCB. On one PCB two step-up/step-down voltage regulators, 5 V and 3.3 V, a MOSFet motor driver, 16 MHz crystal, HDMI mini connector (used to connect to the computer via a programmer) and four LEDs resides. The second PCB houses the tactile buttons and button debounce filters. The authors used the Computer numeric controlled (CNC) milling machine in order to drill all holes and to cut the PCB into the desired shapes before etching them. After the etching procedure was done a solder mask of epoxy was applied to protect the PCB before mounting the components. The result can be viewed below.

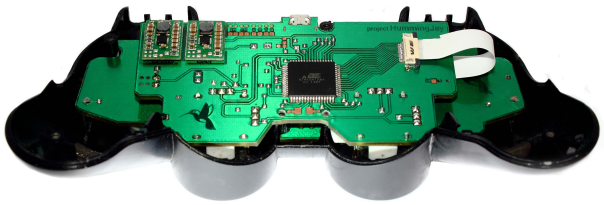

Fig 2.3: Platform controller semi assembled.

As seen in *Fig* 2.4. An extra compartment was added. Within this box the radio transceiver resides, far away from any metallic objects to ensure as good performance as possible.

2.2 Motors

To keep the platform aloft four motors motor/propeller pairs are used. There are many different motor technologies available. For this application there are

Fig 2.4: Complete controller with a mysterious extra compartment.

two different varieties of electric motors to consider, namely permanent magnet synchronous motor (PMSM) and brushed DC motors. A brushless motor have several advantages over the brushed DC motor. See the list below.

Advantages:

- The stator and rotor is galvanically isolated, i.e no brushes.
  - Longer life span.
  - Greater reliability.
  - Reduced electromagnetic interference.
- Greater torque to mass ratio.
  - More torque yields a faster response time for the craft.
- Better efficiency.
  - Increases the time of flight.

But there are of course some disadvantages as well.

Disadvantages:

- The electronics to control the motor is more complex.
  - More components, more weight.
  - Higher cost.

A PMSM, three different propellers and the motor control unit was therefore purchased and evaluated. The small amount of data the manufacturer provided is showed in *Tab* 2.1.

Tab 2.1: Motor characteristics

Model	2210N 1000Kv Brushless Motor
Weight [kg]	0.045
Max Current [A]	11.2
Max Voltage [V]	11

Two of the propellers were 3-bladed and one was two-bladed but all differed in diameter and pitch. The pitch describes the distance, in inches, the propeller would traverse through a soft solid during one revolution. .

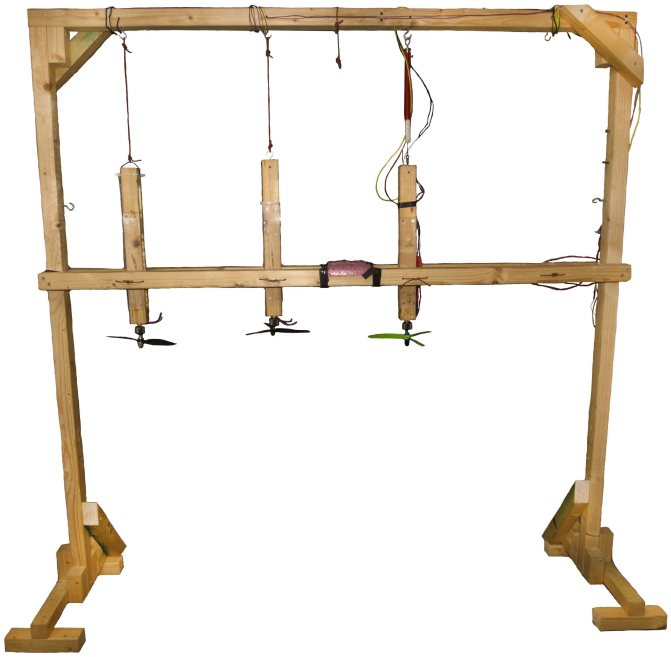

Fig 2.5: Motor and propeller test rig.

In order to set the RPM of the motor a control unit is utilized. The unit produces a three-phase AC output from a DC power supply where the frequency is proportional to the input control signal, which in this case is a PWM signal. By increasing the pulse width the unit increases the frequency, and if necessary the output power. The motors, propellers and the control units were all bought at a hobby store.

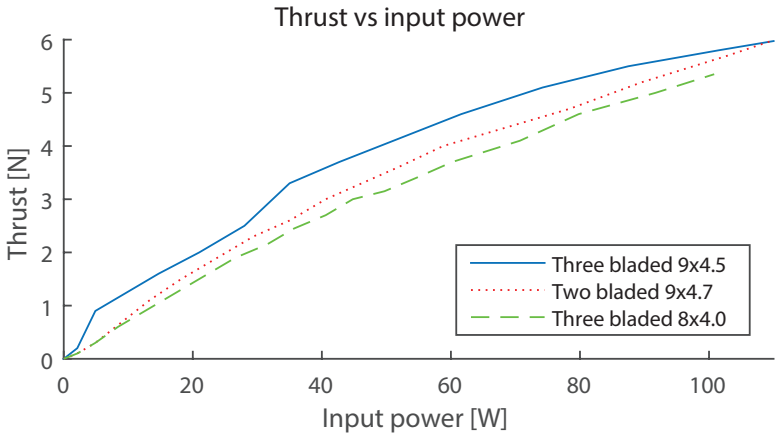

Fig 2.6: Motor and propeller measurement, where V_{motor} is constant at 14 V.

A test rig was constructed in which a motor and propeller attached to a wooden stud that could be hanged up side down via dynamometer, see *Fig* 2.5 below.

Each propeller underwent the same evaluation procedure, which was measuring the input power to the motor control unit and the generated thrust of the motor between 0%-100% of throttle, see *Fig* 2.6.

As seen in the plot the obvious choice is the three-bladed propeller with the diameter and pitch equal to 9 respectively 4.5 inch as it generates the most amount of thrust at any given input power.

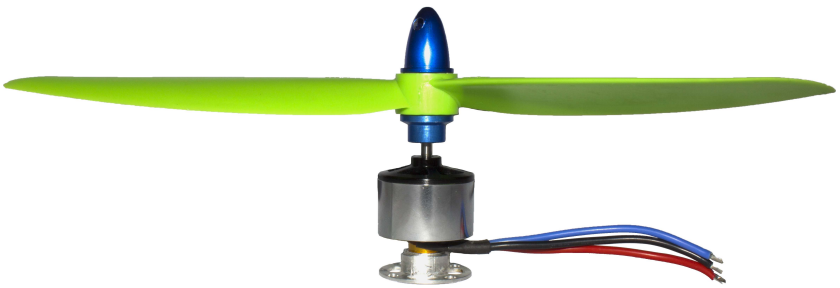

Fig 2.7: The motor and chosen propeller (three-bladed 9x4.5).

One can also draw the conclusion that if four of these propeller/motor pair is combined the maximum sum of the generated thrust, F_{max}, is equal to 24 N. This gives an indication of the maximum allowed mass the craft can posses. As an initial goal this is set to 1.2 kg, which will grant 12 N of an excess of force to reduce the vehicles response time. The motor with the most optimal propeller (Hobbyking 3-Blade Propeller 9x4.5) can be viewed in *Fig* 2.7.

If the mass is indeed 1.2 kg the current consumption at the crafts equilibrium state can be estimated to:

$$\frac{30\text{ W}}{14\text{ V}} \cdot 4 \approx 8.6\text{ A} \tag{2.1}$$

This will be useful while choosing a suitable power supply for the craft.

2.3 Sensors - Pitch and roll

As previously described the angles θ and ϕ is imperative to be estimate with sufficient accuracy, in order to reach stability and perform basic maneuvering in the horizontal plane. There are, to the authors current knowledge, two obvious categories of sensors that could be utilized to estimate this quantity, a gyroscope or an accelerometer. The authors decided that the following requirements has do be met whilst choosing the sensors:

- give a sufficiently accurate estimation of the desired physical quantity,
- small form factor and
- relatively budget-friendly.

As discussed earlier the MEMS senors meet these criteria.

MEMS Gyroscope

A gyroscope, LY33ALH, from ST Microelectronics was chosen. The working principal of this particular sensor is outside of the scope of this thesis. The reader should regard the gyroscope as a "black box" with the following properties:

Tab 2.2: Gyroscope characteristics

Parameter	Value	Unit
Measurement range	± 300	degree/s
Sensitivity	3.75	mV/degree/s
Zero-rate level	1.5	V
Rate noise density	0.014	degree/s/$\sqrt{\text{Hz}}$
Bandwidth	140	Hz

Measurment range

If the craft performs a change in its angular inclination at the maximum angular velocity of $\pm$ 300°/s means that a movement of 45° is performed in 150 ms. A movement faster than this will not be in the linear range of the sensor. Angular velocities of this magnitude is presumed to not happen during a flight.

Sensitivity

A movement of 45° in 1 second seems as an good target. The output signal will have a peak value of 168.75 mV. This is acceptable.

Zero-rate level

The zero-rate level is the quiescent point for the sensor, e.g V_{out} when the angular velocity is equal to zero. For convenience, it would be satisfactory if this level is placed in the vicinity of the center point at the range of the ADC.

Rate noise density

The noise density ratio sets the least detectable signal at any given bandwidth. By decreasing the bandwidth one is able to detect a smaller signal. This is one reason why analog sensors were chosen. Furthermore, for this particular sensor the noise density is relatively low compared to other sensors.

Bandwidth

Angular movement with a period shorter than 7.14 ms is not likely to be a problem due to the mass and size of the quadrotor.

Quiescent Supply Current

The sensor consumes 4.2 mA. This can be neglected when compared to the four motors that consumes over 2000 times more at the crafts theoretical equilibrium state.

As shown in *Tab* 2.2 above the output of the sensor is directly proportional to the angular velocity, which is not ideal and imposes that a mathematical process, a quadrature, has to be performed to get an estimation of the crafts angular inclination. This introduces both some demand on the processing unit and a problem regarding long term drifting which is caused by a couple of imperfections.

First, the sensors zero-rate-bias has to be eliminated before the quadrature can take place. The bias elimination can be implemented in various manners, for instance, with a high-pass filter with an very low corner frequency, but this is of course at a cost of losing some information of angular movement at the lower part of frequency spectrum. Another approach is to simply subtract a constant equal to the bias from the signal. This method is highly dependent on a totally static bias, even the slightest change in sensors bias will yield an angular drift without any rotational movement. It is known that this type of sensors suffers from what is referred to as angular random walk (ARW) which is exactly a small change of the sensor bias. And secondly, in practice, a quadrature can never be performed completely loss-less since it's not analytically. Which can translate into a small accumulating error. A third aggravating circumstance is that there are no perfect data converters, the offset error drift of an analog-to-digital converter may very well contribute to what seems like ARW. To explore to what extent the above mentioned imperfections pollutes the estimation of the angular inclination performed by a gyroscope a microcontroller were used. The authors sampled and accumulated 100 samples from the gyroscope with a resolution of 12-bits at a zero rate level. The mean value of the samples was then calculated and subtracted from every new AD-conversion before the quadrature was performed.

The data was logged by a computer and the result can be viewed in *Fig*. 3.1.

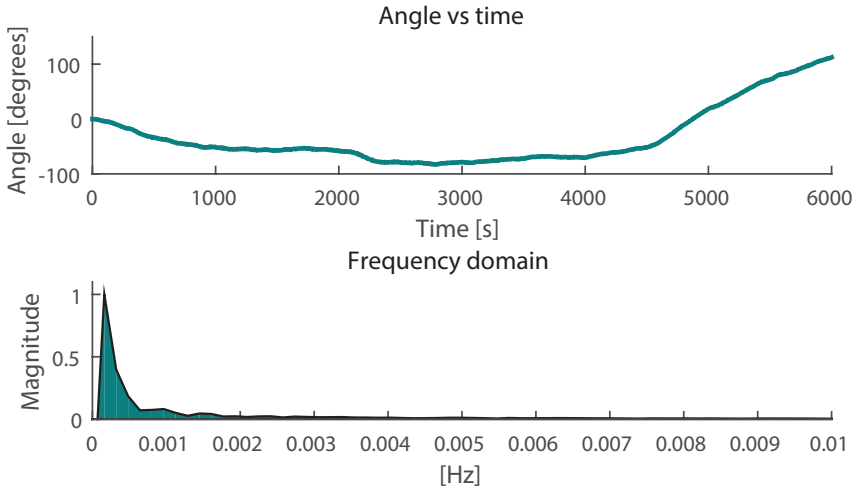

Fig 2.8: Angular drift, 1 hour and 40 minutes.

The quadrature performed was based upon the Trapezoidal rule, please see (2.2).

$$\int_t^{t+h} f(t)dt \approx hf(t) + h\frac{f(t+h) - f(t)}{2} = \frac{h}{2}(f(t) + f(t+h)) \tag{2.2}$$

(The above is of course only one slice of the integral.)

As seen in *Fig*. 2.8 the angular drift is substantial and it is useless if it would be implemented on a flying vehicle as part of the feedback network for the stability control.

The plot of the signal in the frequency domain clearly shows which angular velocities that will get lost due to a poor Signal-to-Noise Ratio (SNR). One has to be aware of the fact that even the smallest deviation from the equilibrium state will make the quadrotor move, hence even the lower part of the frequency spectrum is of importance.

The result can not be explained only by a mismatch of the bias elimination term due to the fact that the derivative of the curve in the top graph in *Fig* 2.8 is changing sign. For example, if the sensor bias change over time in such a way that the signal corresponds to the one depicted in *Fig* 2.9 the result of the quadrature will be a close match to the what is shown in *Fig* 2.8. Furthermore the ADC offset error drift could also explain this.

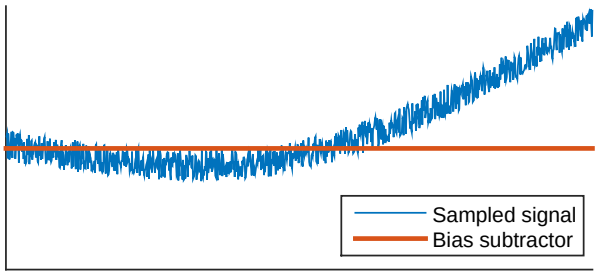

Fig 2.9: Sensor bias change.

The careful reader will most likely have noticed that the time elapsed during the measurement is 1 hour and 40 minutes which is well beyond the vehicles time of flight (TOF), but after only 3 minutes, which is well within the TOF, it has drifted almost 10 °. See *Fig* 2.10.

After this examination one can draw the conclusion that the gyroscope is not an optimal sensor for this application since it will only be able to provide information that lies outside of the contaminated frequency interval. The other option is

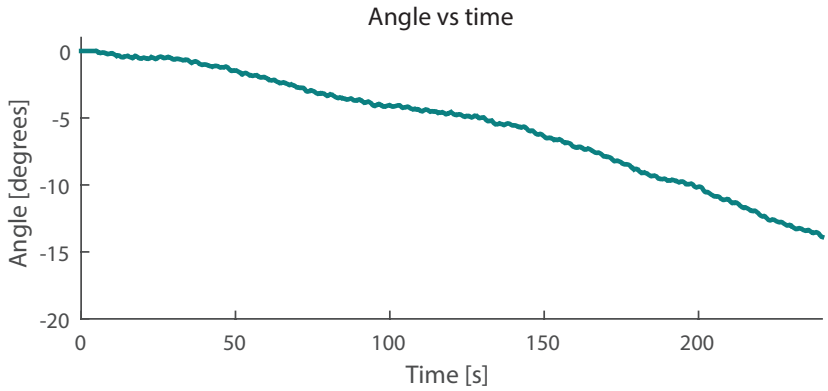

Fig 2.10: Angular drift, 3 minutes.

to use an accelerometer. This will be discussed in the following section.

MEMS Accelerometer

As one uses a accelerometer to estimate the angular inclination it is necessary to utilize the gravitational field as a reference. While the sensor is place perpendicular to the field lines, and not moving, the acceleration component that is parallel with the horizontal plane is zero but as it the sensor is rotated it is not, see *Fig* 2.11.

The output voltage from the sensor can therefore be described by (2.3)

$$V_{out} = \sin(\phi) \cdot S \tag{2.3}$$

Where S is the sensor sensitivity ([V/g]).

But it is possible to make a drastic simplification by considering the Taylor series of $\sin(x)$.

$$\sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{1+2k}}{(1+2k)!} = \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} + ... \tag{2.4}$$

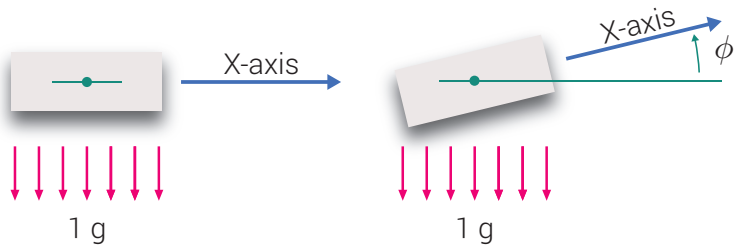

Fig 2.11: Accelerometer and inclination sensing.

By looking at the expression above (more particularly the terms exponents) one can come to the conclusion that

$$\sin(\phi) \approx \phi \text{ for small } \phi \tag{2.5}$$

and (2.3) can be simplified to

$$\phi \approx \frac{V_{out}}{S} \tag{2.6}$$

This heavily reduces (2.3) and makes computational effort negligible (trigonometric functions are demanding to evaluate, especially in a microcontroller and cumbersome to implement with analog circuits).

The error of the approximation is $\sin(\phi) - x$ and is shown in *Fig* 2.12 and is roughly 2% at 20° which is an acceptable error at this point. Furthermore 20° will be set to the maximum allowed angular inclination of the platform. An approximation of the resulting acceleration as a function of the angular inclination will be given in the section regarding the inertial navigation.

Two accelerometers, ADXL335 and ADXL203CE, were chosen to be further investigated. The working principle of these particular sensors will be proven to be a useful knowledge for the reader later on so a brief description will be given.

As seen in *Fig* 2.13 the sensor essentially consists of a seismic mass held by a system of springs-like structures. On the mass a conductive plate is positioned and fitted in between two other conductive plates which are fixed to the casing. Together the plates forms two capacitors. While the mass is not subjected to an

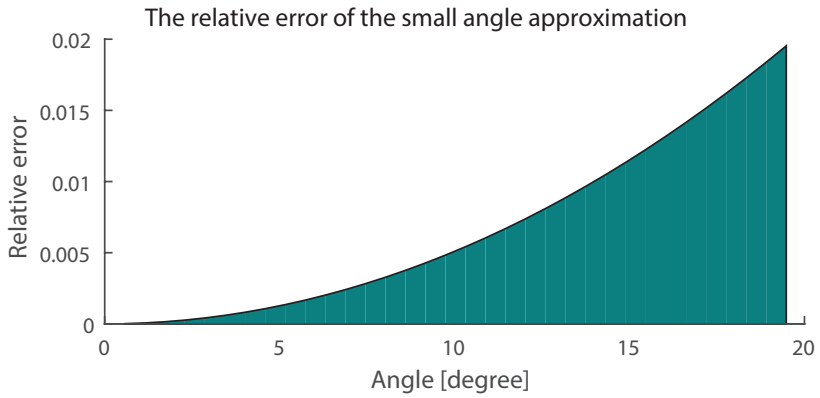

Fig 2.12: The absolut error of the angle approximation.

acceleration the capacitance of the two capacitors is equal, but as soon as it is exposed to an acceleration, there will be a difference.

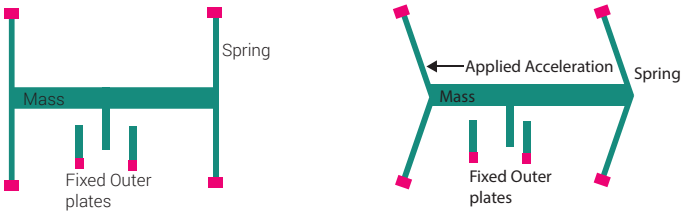

Fig 2.13: Mems Acceleometer

The difference in capacitance is then translated into a voltage. In the case of the accelerometers here chosen, the output voltage is in direct proportion to the applied acceleration.

In *Tab* 3.2 and 3.3 some of the interesting characteristics of the accelerometers could be found.

Tab 2.3: ADXL203CE characteristics @ 5 V

Parameter	Value	Unit
Measurement Range	$\pm$ 1.7	g
Sensitivity at X_{out}, Y_{out}	1000	mV/g
0 g Voltage at X_{out}, Y_{out}	2.5	V
Noise Density	110	$\mu g/\sqrt{Hz}$
Sensor Resonant Frequency	5.5	kHz
Operating Voltage Range	3-6	V
Quiescent Supply Current	0.7	mA

Tab 2.4: ADXL335 characteristics @ 3.6 V

Parameter	Value	Unit
Measurement Range	$\pm$ 3	g
Sensitivity at X_{out}, Y_{out}, Z_{out}	300	mV/g
0 g Voltage at X_{out}, Y_{out}, Z_{out}	1.5	V
Noise Density X_{out}, Y_{out}	150	$\mu g/\sqrt{Hz}$
Noise Density at Z_{out}	300	$\mu g/\sqrt{Hz}$
Bandwidth X_{out}, Y_{out}	1600	Hz
Bandwidth Z_{out}	550	Hz
Sensor Resonant Frequency	5.5	kHz
Operating Voltage Range	1.8-3.6	V
Quiescent Supply Current	350	μA

The ADXL203CE seems particularly promising due to the its high sensitivity and low noise density, this is of course reflected in it's price. One should also be aware of that the output signal is ratiometric, i.e. the output is directly proportional to the supply voltage.

Measurment range

In order to approximate the angle from 0°-90° the needed measurement range has to be 1 g. Thus both of the sensor will be acceptable.

Sensitivity

The higher sensitivity option is, by natural reasons, the most desirable.

Zero g Voltage

The zero-rate level is the quiescent point for the sensor, i.e V_{out} when the acceleration is equal to zero. As for the gyroscope, it would be convenient

if this level is placed in the vicinity of the center point at the input range of the ADC, which in both case are true.

Noise Density

The ADXL203CE is superior since the noise density is lower than for the ADXL335.

Bandwidth

The expected accelerations is estimated to be within both sensors bandwidth.

Sensor Resonant Frequency

This could pose a problem due to the fact that the the motors revolutions per minute range spans over the sensor resonance frequency. The highest RPM of a motor/propeller pair was measured to 6700.

Operating Voltage Range

Since the sensors are of the ratiometric type it is very important that the operating voltage is fixed. Voltage regulators with sufficiently good line regulation and ripple rejections has to be used.

Quiescent Supply Current

The sensor consumes 0.35 mA, which is, as for the gyroscope, negligible compared to the motors.

While measuring the objects angular inclination with an accelerometer the sensor will never measure an acceleration that exceeds 1 g. However, this presupposes that the the vehicle is not subjected to any other accelerations in the sensors sensitive direction. The acceleration measured at the maximum angle is equivalent to 0.34 g. By choosing the ADXL203CE, which has a measurement range of $\pm$ 1.7 g one is granted the best available resolution while still having a margin.

To evaluate the accelerometer, it was soldered on a small "break-out"-PCB and mounted on a breadboard, an external power supply connected to a voltage regulator (LM317) which supplied the power to the sensor. In the digital domain the signal was divided by the sensor sensitivity and then inverse sine value was computed in order to calculate the breadboards angular inclination.

Unfortunately there was no time to build a proper automated test rig for the sensors so the authors had to improvise. The breadboard was tilted back and forth

between 0°-45° by hand and the result can be viewed in *Fig* 2.14 below.

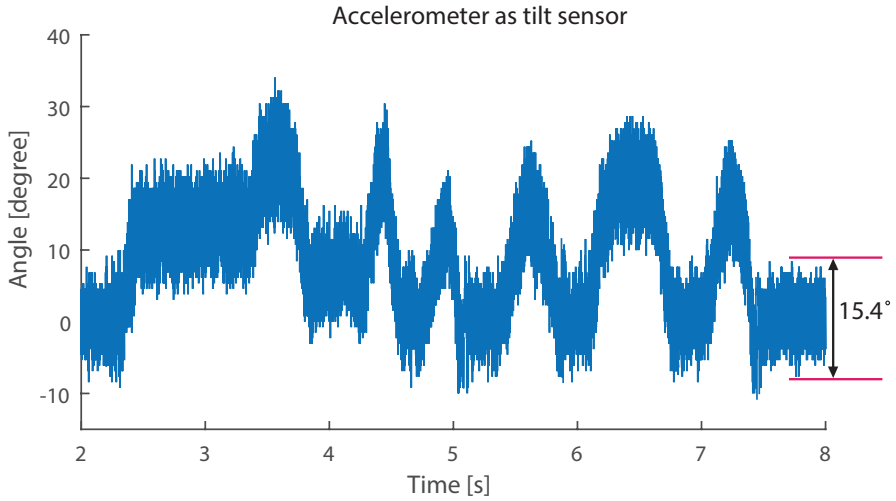

Fig 2.14: Sampled with an oscilloscope with a sampling frequency of 12.5 kHz. The small angle approximation was not used.

As seen the sensors suffers form an undesirable traits which yields a noisy output. At rest the peak-to-peak value of the noise is more than 15°, this of course poses a problem. An acceleration that gives a signal below the noise floor can simply not be measured accurately. This problem can be prevented by reducing the bandwidth of the sensor which will improve the SNR. By using the manufacturer provided data and equation (2.7) below one can calculate the least detectable acceleration.

$$a_{least} = n \cdot \sqrt{B} \tag{2.7}$$

Where n is the noise density ratio and B is the signal bandwidth.

An analog low pass filter was therefore introduced and the signal was sampled (f_s = 12.5 kHz) before and after the filter. The result can be seen in *Fig* 2.15. According to (2.7) it should now be possible to detect a signal that corresponds to 82 μg. One may think that the reduction of the bandwidth is too drastic but there is in fact another reason for restricting the bandwidth while using the accelerometer to estimate the angular inclination of a moving object like a quadrotor platform. Whilst the craft is manoeuvring and the movement perpendicular with the sensitive axis of the sensor, that movement will give rise to an acceleration that will be measured and interpreted as if the craft was tilted. This is of course an unwanted side effect and has to be suppressed as much as possible. One way to achieve

this is to drastically decrease the bandwidth of the output from the accelerometer. After a series of trial based experiments a suitable bandwidth of the signal was determined to be around 0.3 Hz.

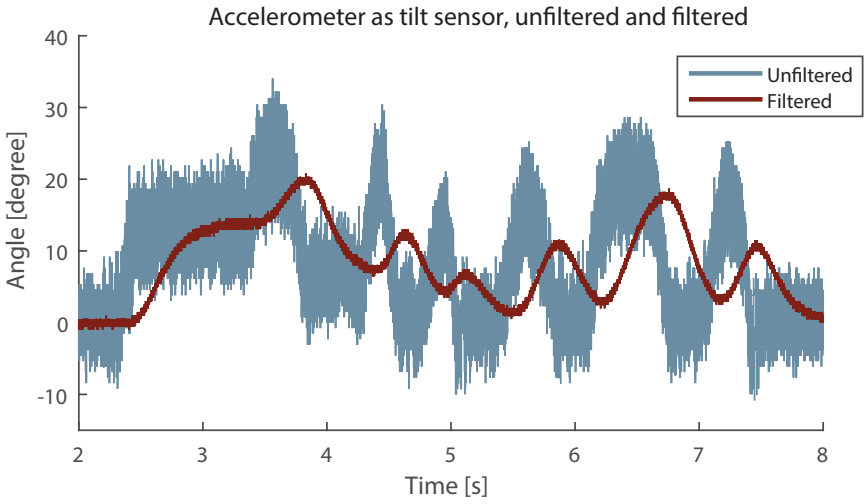

Fig 2.15: Filtered and unfiltered accelerometer data, Sampled with an oscilloscope with a sampling frequency of 12.5 kHz. The corner frequency for the low pass filter is 0.3 Hz. The small angle approximation was not used.

A hard truth is that every improvement comes with a sacrifice, which in this case is that by reducing the bandwidth the dynamic range of the sensor will be limited. Angular change with frequencies higher than 0.3 Hz will be suppressed and delayed. This renders the sensor useless for the upper part of the frequency spectrum.

Solution

The previously described sensors suffers from unwanted characteristics but at different parts of the frequency spectrum. A possible solution would be to high pass filter the gyroscope signal to eliminate the bias drifting, before computing the integral, and to low pass filter the accelerometer signal to suppress noise and acceleration readings that arises from fast lateral movement of the platform. In this way the sensor signals will complement each other and if both filters are matched the circuit should have a flat (flat enough for the application) frequency response. See *Fig* (2.16) for a graphical reference.

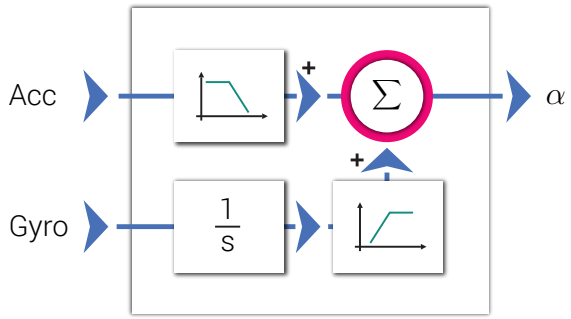

Fig 2.16: A summation of the accelerometer and the integral of the gyroscope signal.

2.4 Sensors - Yaw axis

In the case of estimating the angular position about the yaw axis it is not possible to rely on an accelerometer for the lower part of the frequency spectrum. Because of this reason a sensor that utilizes a different external reference is needed. The earth's geomagnetic field's horizontal component is selected as reference and a sensor with sufficient sensitivity is required for this. The field strength of the horizontal component B_h, which is to be measured, has a size of about 25 μT. The relationship between the direction of the field and the output is described by a sinusoidal signal.

A gyro sensor, the same type as for the θ or ϕ angles, a hall effect sensor of ratiometric type and a complementary filter, as for the roll and yaw-axis, was selected for the purpose.

Hall effect sensor

Every value of the output signal from the hall effect sensor corresponding to two angular positions during a revolution, 360°. By combining the two sensors with a mutual offset of 90°, the direction can be explicitly determined by the unique two-dimensional output signal see *Fig* 2.17.

To increase the resolution of the measurement ranges and to linearize the relationship with an error of less than 4% as in *Fig* 2.18,

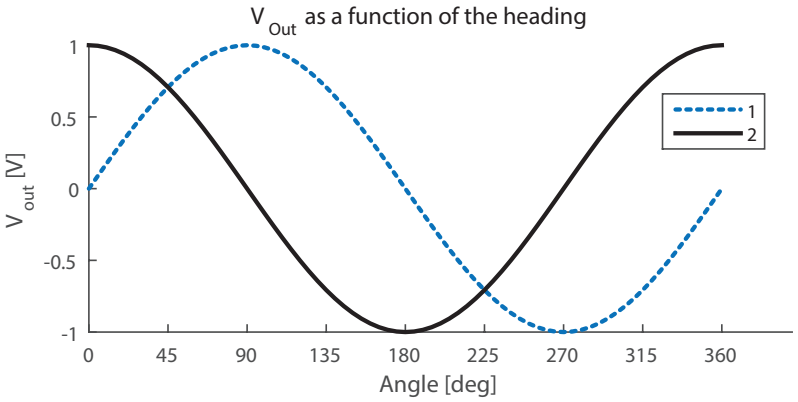

Fig 2.17: A unique two-dimensional output.

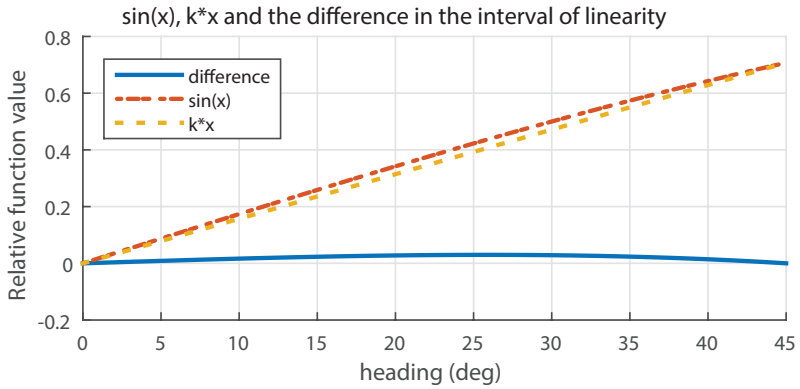

Fig 2.18: Error of the non-linearity.

the authors used selected parts of the output curves, namely those located in the area of $\pm\, 1/\sqrt{2}$ of the signal amplitude, see *Fig* 2.19.
By means of the described curve sections, a continuous signal could be created, linearly proportional to the angle ϕ according to the following

$$A : \text{if curve } 1 > V_1 \text{ with the value of } v_a, \tag{2.8}$$
$$B : \text{if curve } 2 < V_2 \text{ with the value of } v_b, \tag{2.9}$$
$$C : \text{if curve } 1 < V_2 \text{ with the value of } v_c, \tag{2.10}$$
$$D : \text{if curve } 2 > V_1 \text{ with the value of } v_d, \tag{2.11}$$

where the integer index of v denotes the signal (*Fig* 2.17) and V_1 and V_2 is the upper and lower voltage limits, see *Fig* 2.20.

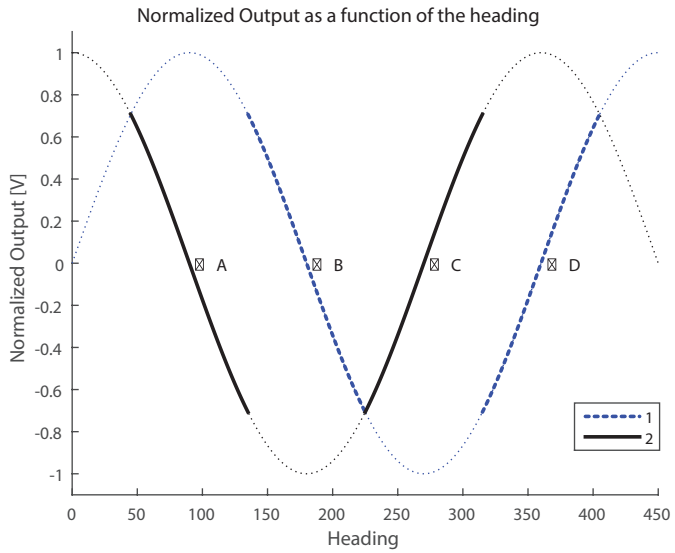

Fig 2.19: Selected portions of the output signal.

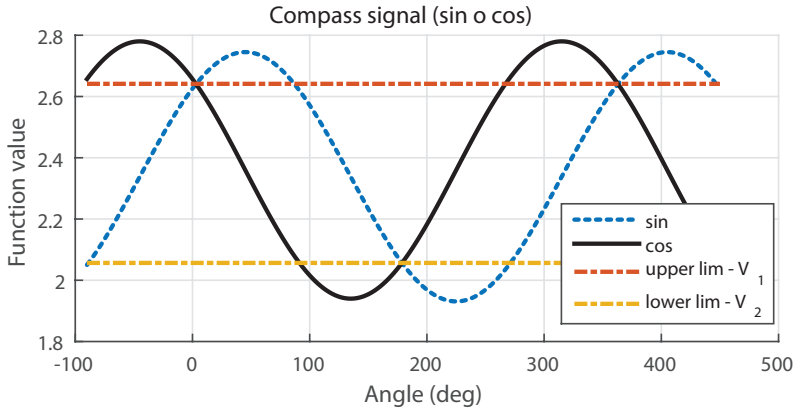

Fig 2.20: Selected portions of the output signal with amplitude limits marked.

The compass output signal can now be constructed as

$$\begin{aligned} V_C = & V_1 - v_a, \\ & V_1 - v_b + (V_1 - V_2), \\ & -V_2 + v_c + 2 \cdot (V_1 - V_2), \\ & -V_2 + v_d + 3 \cdot (V_1 - V_2) \end{aligned}$$

for each interval, A, B, C and D.

Due to the fact that the amplitude of signal 1 and 2 are not exactly equal, V1 had to be decreased and V2 slightly raised. As a bi-product an overlap was created. Inside the overlap a mean value of the two selected curves was created. New reference values were made where

$$\begin{aligned} V_1 &\rightarrow V_a \\ V_2 &\rightarrow V_b \\ V_3 &\rightarrow V_c \\ V_4 &\rightarrow V_d \end{aligned}$$

which also were made adjustable so that the curve portions could be matched in the joints. V1 and V2 thus became references for selecting the usable part of the curve V_a, V_b, V_c and V_d became the basis for the outputs and gave V_C.

Furthermore, this module will be referred to as the compass, by obvious reasons, in the forthcoming chapters.

2.5 Communication link

A communication link between the operator and the quadrotor platform is required. Via this link flight instructions will be sent. For this purpose a XBee PRO S1 was used.

The XBee is an RF module that employs the IEEE 802.15.4 network protocol which allows the device to conduct peer-to-peer, point-to-point and point-to-multipoint communication at a reasonably high transfer rate, namely 250 kbit/s. The module is designed for applications where low power consumption and low latency is requested. The output power is 63 mW which yields an indoor range of 90 meters and for outdoor and in line-of-sight up to 1.6 km. The device is interfaced via the de facto serial communication standard UART (Universal Asynchronous Receiver/Transmitter) and supports an interface data rate equivalent to up to 115.2 kbit/s. This meets the requirements of the authors supposed application. The module can be configured in numerous ways but it was decided to minimize the complexity. There were only few changes made from the provided settings. The device was initialized in such a way that the latency was minimized, this was partly achieved by not allowing it re-send lost data packets and to not wait for packet acknowledges.

2.6 Power Supply

A truly vital part of a the quadrotor is the power supply. A four cell lithium polymer battery with the capacity of 4000 mAh and rated to be able to delivery a constant discharge current of 100 A was chosen for this purpose. As concluded earlier the current consumption at the crafts equilibrium state is approximatively 8.6 A which will yield 30 minuets if hovering time.

2.7 Quadrotor Processing Unit

The quadrotor is an intricate construction mainly consisting of four motors and one motor control unit for each one of them, 2 sensor per axis and radio link. All of these peripheral units needs constant maintenance and surveillance. A central hub to gather and process sensor data and give appropriate commands to actuators is therefore needed. Below follows a short list of required peripheral units.

Timer/Counter

A timer/counter for creating PWM-signals will be necessary. It will be used to interface the motor control unit.

Universal Asynchronous Receiver/Transmitter

As previously described, the radio transceiver is interfaced via UART

Data conversion

The chosen sensors has an analog output. Therefore it is needed to convert the signal into the digital domain. A more detailed specification of the required resolution will follow in the next section.

Analog-to-Digital conversion - Pitch, roll and yaw

To determine what resolution is needed for the data conversion it is important to get a sense of what angular precision that is required to maintain in the equilibrium state. If a resolution of 12-bit is chosen the least significant bit would be equivalent 0.81 mV or 0.0079 m/s^2 and this would yield angular resolution of 0.05° (if the signal from the complementary filter is not amplified) and a lateral drift of 0.2 m in 10 s. This is more than enough, but this is of course entirely theoretical.

The compass has a theoretical resolution of 7 $mV/^\circ$ which corresponds to roughly 8.7 $qu/^\circ$ (qu - quantization unit, the value of the LSB) at a resolution of 12-bits. This is acceptable due to the fact that the yaw direction is of less importance related to the angles θ and ϕ.

Computational cost

The motor control unit can be updated with a maximum frequency at 400 Hz which implies that all of the control algorithms has to be computed within 2.5 ms. This has to be kept in mind while choosing the processing unit.

Processing Unit - Chosen

After careful considerations a microcontroller (TMS320F2806F28069 - Piccolo) manufactured by Texas Instrument was selected, and our friends at the local office were kind enough to sponsor the project and supplied the authors with two development kits. The development kit consisted of the microcontroller, a development board and JTAG (programming and debugging interface).

The microcontroller has two cores, the main core is a 32-bit CPU with a clock frequency of 90 MHz. Some of the peripheral units that are available and that will be partly used here are the following:

16 PWM channels, each with an independent timer
- Four of them will be used to control the motor unit.

Three 32-bit CPU timers
- One of them will be used to trigger the ADC, for the control algorithms and for a possible inertial navigation system.

Two serial communication interfaces (UART)
- One is utilized to send and receive command via the radio link to the remote controller.

16 12-bit ADC channels
- 3 channels are occupied by the complementary filter and some of the remaining will be allocated by accelerometers dedicated for the inertial navigation system.

One floating-point unit
- Speed up calculations with a need of more accuracy.

54 multiplexed general purpose input/output pins
- LED's and miscellaneous.

100KB RAM and 256 KB Flash
- It is always good with some memory.

The above is not the only reason that made the authors chose this particular microcontroller. The Piccolo MCU have a another compelling feature and that is the other core, the Control Law Accelerator, which will be described in the subsequent section.

Control Law Accelerator

The Control Law Accelerator (CLA) is a 32-bit floating-point math accelerator which operates independently of the MCU's main CPU. It has an independent register set, memory, bus structure and processing unit. The CLA has direct access to the PWM-module and the ADC result register. It is possible for the CLA to read the ADC result registers in the same cycle as it updates which reduces the sample-to-output delay.
The CLA has 8 programmable tasks, which essentially are simplified interrupt services routines. They can be triggered by a PWM-interrupt (at the end of the PWM period) or when an ADC conversion is completed and from software. The tasks are prioritized, task 1 has the highest priority while task 8 has the lowest. This means that if two task are called at once the task with the highest priority is executed first. There is no possibility to perform any type of nested tasks, i.e. a task can not call another task. There are neither any conventional interrupt service routines available, which shortens the response time when a task is invoked since a task is always is completed before another is executed.

The Control Law Accelerator has its own assembly instruction set but Texas Instruments is also providing a C compiler, but the manufacturer highly recommends to use the assembly instruction set and the authors acted upon that advice, since it will be more convenient in regards to keep track of the tasks execution time.

The CLA and the main CPU can pass data between each other via a message RAM block. The block is divided into two parts, writes from the CPU and reads from the CLA is allowed for one and writes from the CLA and reads from the CPU in the other.

The CLA is fully compatible with the IEEE 32-bit floating-point format (single precision floating-point). All basic operations such as add, subtract, multiply, 1/x, move, logic operations, read, store etc is executed in one clock cycle. The mathematical operations is done with full accuracy except for the inverse of x, this is a reciprocal approximation and the result is only accurate to 8-bits. To increase the accuracy to 32-bits the Newton-Raphson algorithm has to be performed twice.

Because of the above the CLA will be used to unload the main CPU by performing the heavier computations, such as the algorithms for the stability control and the inertial navigation. The main cores primary assignments are therefore maintenances related, i.e initializing peripherals and respond to interrupts from UART module and relay the commands to the CLA.

Estimation of execution time

Now it is possible to estimate the execution time for the control algorithms. The clock frequency of the CLA is 90 Mhz thus the available number of clock cycles are 225 000 (because of the motor control units bandwidth) hence it is safe to say that there is a significant margin. The number of mathematical operations a simple PID controller has to perform is 9, which is equal to the number of needed clock cycles the execute them, if divisions is avoided and replaced with multiplications.

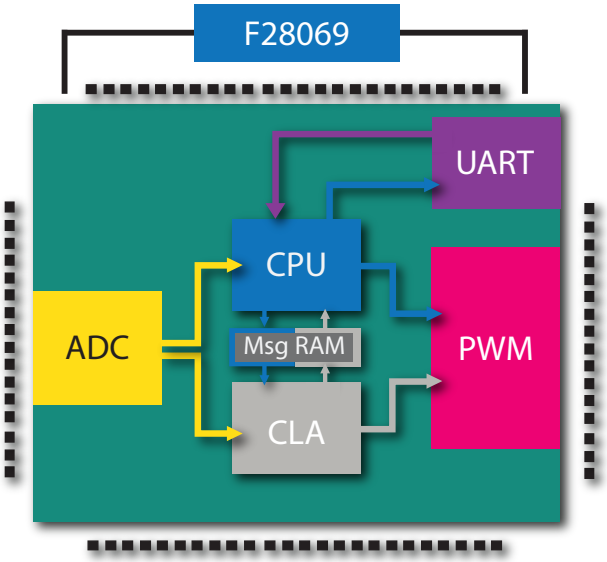

Fig 2.21: Piccolo, simplified core structure.

2.8 Software

As explained before the main CPU is mostly utilized to perform maintenance related task. It will handle the initialization of peripheral units such (serial communication interface, the PWM unit, the analog-to-digital converter and the control law accelerator).

After the initialization sequence is performed the CPU's main duty will be to receive commands from the remote controller and send them via the message RAM to CLA as target values for the control algorithms. It will also handle the trimming process (which is invoked and carried out by the operator) of some variable values in order to reach the crafts equilibrium state, i.e. hovering.

A more detailed description of the control algorithm the CLA performs will be given in the next section.

Stability control

As previously described the platform is inherently unstable and is in need of a stability control system. The chosen control structure is based on the work of G. Szafranski, R. Czyba. *Different Approaches of PID Control UAV Type Quadrotor* [5] The system closed loop transfer function in block form is showed in *Fig* 2.22. As

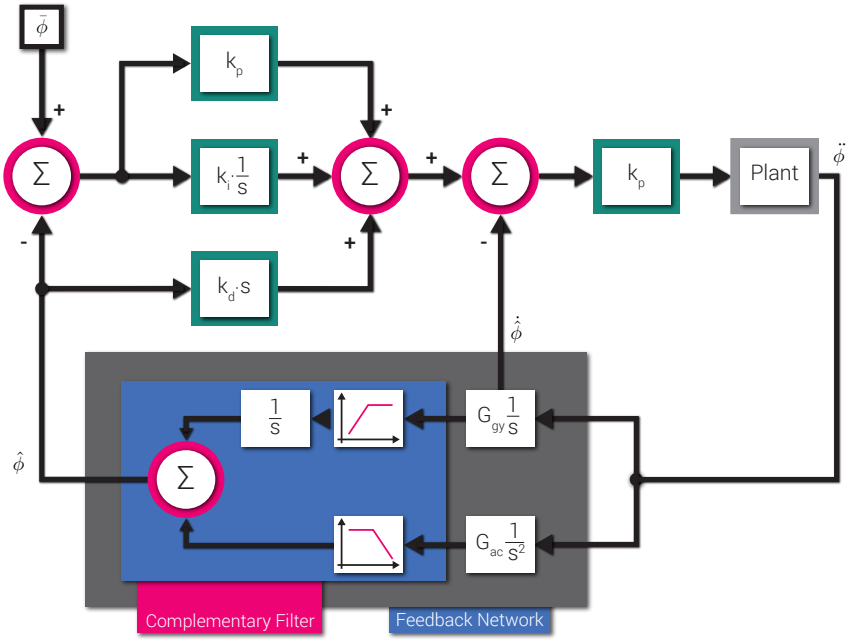

Fig 2.22: System transfer function in block form.

seen in the figure a cascaded control loop used. The inner loop will handle the higher frequency dynamics and attenuates the plants non-linearities while the outer loop controller produces the target value for the inner.

The anglular velocity is used as a reference value for the inner controller since it will give information about the system faster than the signal that estimates the angle (derivative of angle = angular velocity). As one may remember the signal from the gyroscope is biased and this is of course a problem. The bias has to be eliminated before it can be used as an input signal to a control. The elimination procedure could be realized by a high pass filter. This is possible since the inner loop only handles the faster dynamics of the plant and the information loss at frequencies close to zero is of less importance.

An infinite impulse response (IIR) filter is the best choice in this particular case due to its simplicity. The frequency response of this type of filters can be described as the product of the distance from each pole to $e^{j\omega}$ divided the product of the distance from each zero to $e^{j\omega}$ in the z-plane, see equation (2.12) and *Fig* 2.23.

$$\left| H(e^{j\omega}) \right| = \frac{\prod_{k=0}^{m} \left| e^{j\omega} - z_m \right|}{\prod_{k=0}^{n} \left| e^{j\omega} - p_n \right|} \tag{2.12}$$

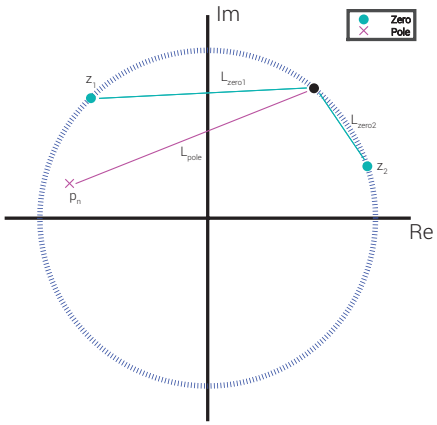

Fig 2.23: Filter output and the relationship between the poles and zeroes.

Thus the conclusion can be summarized to that zeros attenuate and poles amplify. By placing the pole and zero in accordance with *Fig* 2.24 the desired response is obtained.

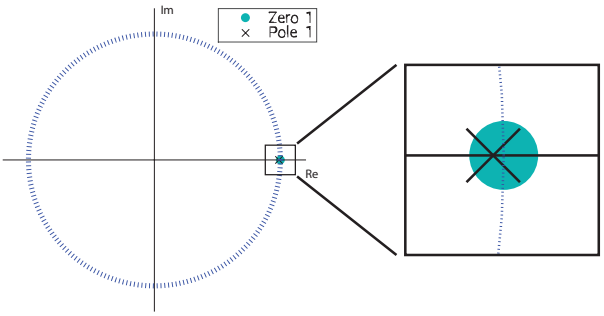

Fig 2.24: Placement of the poles and zeros.

The transfer function is given by (2.13).

$$H(z) = \frac{1 - z^{-1}}{1 - \alpha \cdot z^{-1}} \tag{2.13}$$

Which yields the following expression,

$$Y(z) \cdot (1 - \alpha \cdot z^{-1}) = X(z) \cdot (1 - z^{-1}) \tag{2.14}$$

and in the sampled time domain equation 2.15 can be described with the difference equation below.

$$y_n = x_n - x_{n-1} + \alpha \cdot y_{n-1} \tag{2.15}$$

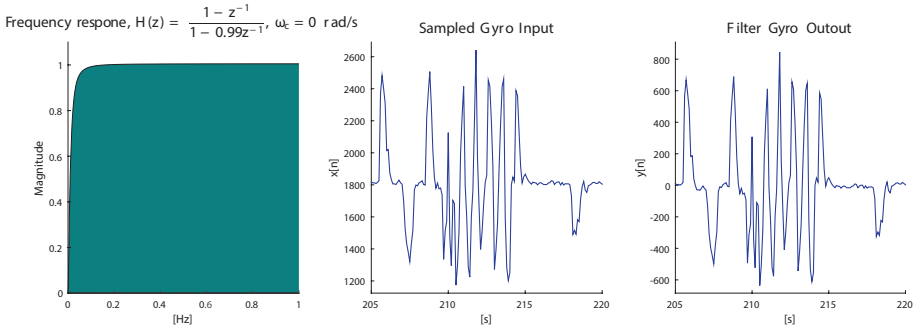

Fig 2.25: Frequency response, sampled gyro as input and the resulted output.

This will be executed every time the CLA is invoked. For a frequency response and a sampled gyroscope signal filtered and un-filterd see *Fig* 2.25.

Furthermore, all the samples from the ADC were low pass filtered. This was done with a finite impulse response (FIR) filter. Such a filter can be derived from the inverse Fourier transform of the ideal frequency response, see equation (2.16) and *Fig* 2.26.

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\omega}) e^{j\omega n} d\omega \tag{2.16}$$

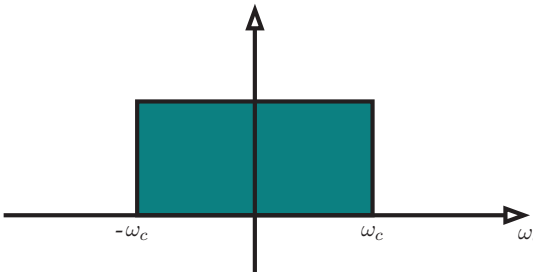

Fig 2.26: Ideal frequency response of a low pass filter.

The ideal filter can thus be described by (2.17) below

$$H(e^{j\omega}) = \begin{cases} 1, & -\omega_c \leq \omega \leq \omega_c \\ 0, & \text{Otherwise} \end{cases} \tag{2.17}$$

which will make the transform easier to solve. See below.

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\frac{1}{jn}\left[e^{j\omega n}\right]_{-\omega_c}^{\omega_c} = \frac{\sin(\omega_c n)}{\pi n} \tag{2.18}$$

The given unit sample response is non-causal and has an infinite duration, thus it must be made causal and the duration has to be finite before it can be implemented. The first problem can be solved by shifting $h(n)$ into positive time and the second by truncating it. The truncation is equivalent to multiplying the response with a window function like the one in (2.19) below,

$$w(\omega) = \begin{cases} 1, & \text{for n = 0 to M-1} \\ 0, & \text{Otherwise} \end{cases} \tag{2.19}$$

where M is the length of the filter.

A rectangular window like (2.19) has a frequency response given by the Fourier transform of $w(\omega)$, see below.

$$W(\omega) = \mathrm{e}^{-j\omega(\text{M-1}/2)} \cdot \frac{\sin(\omega \text{M}/2)}{\sin(\omega/2)} \tag{2.20}$$

By performing this action one introduces unwanted characteristics. It will cause ripples in the stopband, further more the roll-off will be less steep. The ripples originates from the discontinuity of the window but they can be suppressed by using a "more" continuous window. But as in majority of cases, a improvement comes with a trade off, by better the stopband attenuation the filter roll-off worsen. See *Fig* 2.27.

The filter above is of the 24^{th} order, the sampling frequency is 400 Hz. In this case the authors opted for the filter with the better stopband attenuation. The output of the filter can be represented by the non-recursive difference equation which is a linear combination of M-1 individually weighted previous inputs and the current weighted value. See (2.21). In this representation the filter is easily implemented in assembly code.

$$y(n) = \sum_{n=0}^{\text{M}-1} h(n) \cdot x(k-n) \tag{2.21}$$

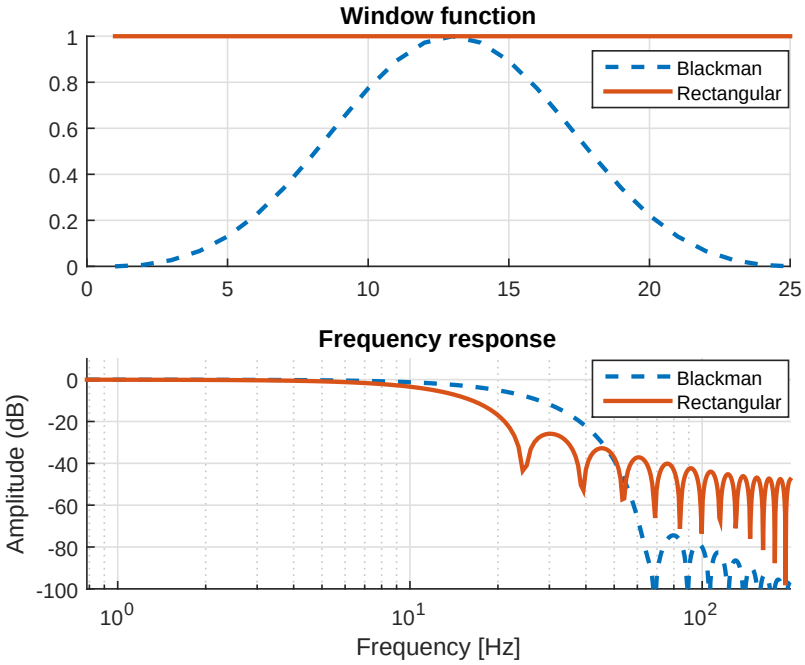

Fig 2.27: Rectangular window vs Blackman.

With the signals in order one can move on to the next topic which is the control algorithm, but before implementing the previously described controller it is good to be aware of two of potential caveats. One has its origin in the derivative path and is known as the derivative kick and the other in the integral path and is called integral wind-up.

Derivative kick

The contribution from the derivative path is shown in equation (2.22) below. This is the derivative of the error signal which is the difference between the target value and the process value.

$$\frac{d}{dt}e(t) = \frac{d}{dt}r(t) - \frac{d}{dt}y(t) \tag{2.22}$$

If there were to be a momentaneous change of the target value the derivative of the error would be a spike. This could have unwanted side effects (for this application it will make the motor change its angular velocity very fast which

could produce vibration) but it is easily prevented. The derivative of the setpoint is most of the time zero, in relation to the evaluation period of the algorithm. Because of this fact it can be omitted and equation (2.22) can be re-written as in (2.23). This is also visible 2.22.

$$\frac{d}{dt}e(t) = -\frac{d}{dt}y(t) \tag{2.23}$$

Integral wind up

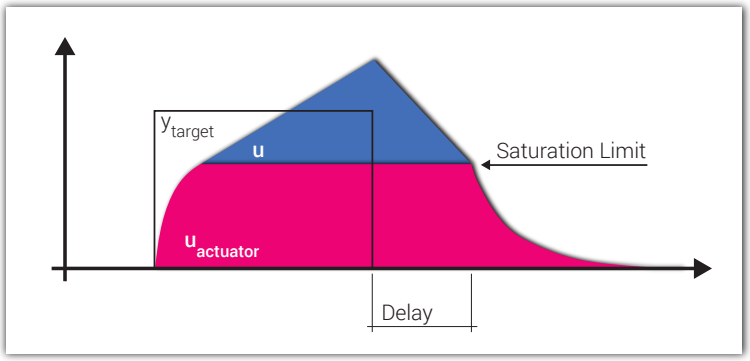

Fig 2.28: Wind up.

A wind up occurs when the actuators becomes fully saturated while the output from the control algorithm still increases. This could happen if the system is forced into a state where the target value is not reached, i.e an error will be present. The integral part will thus continue to grow well past the actuators saturation limit until the error signal change sign. The situation described will introduce an unwanted delay where the system seems unresponsive and shows non linear traits. Please see *Fig* 2.28 for a graphical.

The way the authors chose to prevent this phenomena was simply to never allow the absolute value of the integral part to exceed a predefined maximum limit. The same clamping procedure was also applied on the overall output of the PID controller.

Implementation of the control algorithm

As stated before the control algorithms is carried out by the CLA. The CLA is set to be triggered every time a data conversion is completed, $F_s = 400$ Hz. By

executing the control algorithm at a frequency of 400 Hz one yields a refresh of the motor control signal every 2.5 ms, this is within the specified bandwidth of the control unit. The execution time for all of the CLA's instructions is roughly 10.6 us, which grants some elbow room.

Before the CLA starts evaluating the control algorithm it is set to accumulate 1000 samples from all three complementary filters (individually) and then the mean value is calculated. The mean value is used to eliminate the sensor "zero value" output at each task call. A code execution flowchart is showed in *Fig* 2.29.

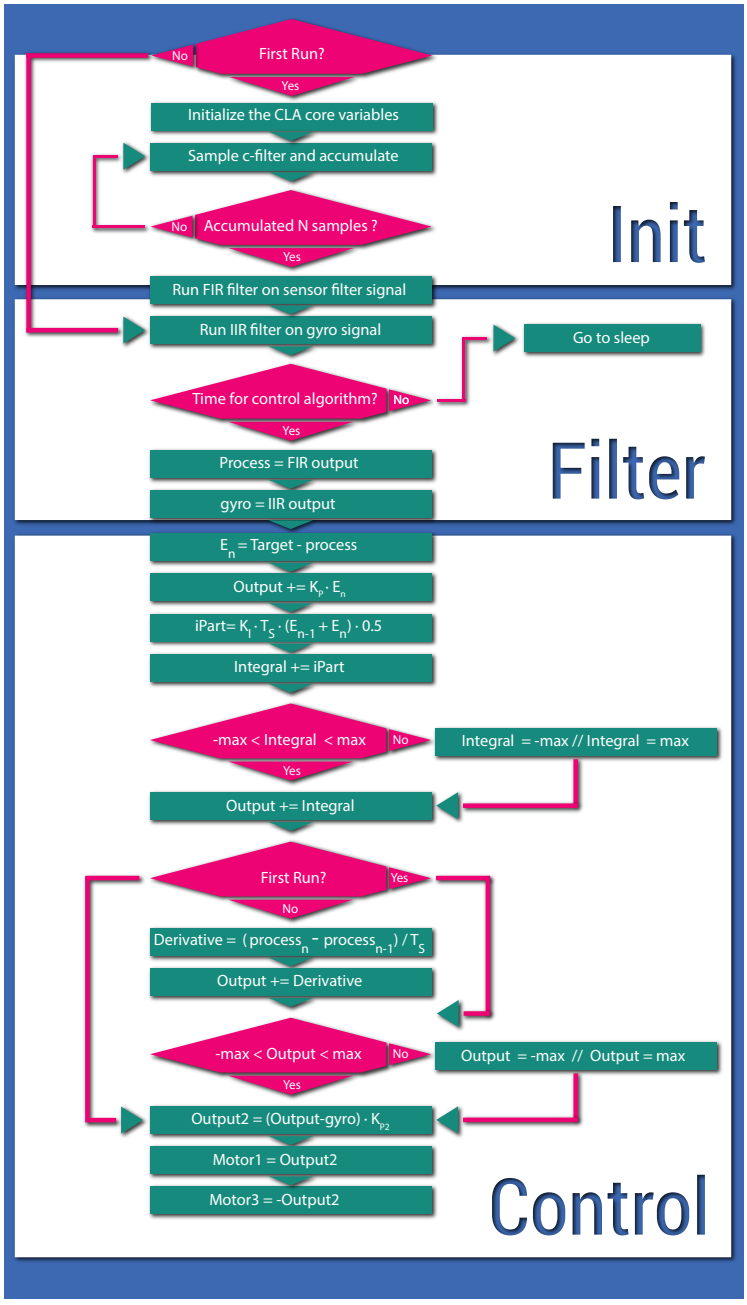

Fig 2.29: Flow chart of the control algorithm.

2.9 Cross frame

As stated in chapter 1.1 a cross frame is needed to keep the essential parts of a quadrotor platform (motors, motor control units, sensors, the processing unit etc) together. When it came to the electronics, the authors decided to strive to achieve a as modular design as possible by compartmentalize the platforms essential functionalities into replaceable blocks in form of smaller PCBs. These PCBs will be connected together on a larger. By doing this the complexity is reduced and broken parts can easily be replaced. The larger board, niftily called the motherboard, will also be a part of the cross frame construction. The connector between the function blocks and the motherboard are of the DIM100 type. The connector has 100 pins/poles, this is more than enough and grants some elbow room for future improvements.

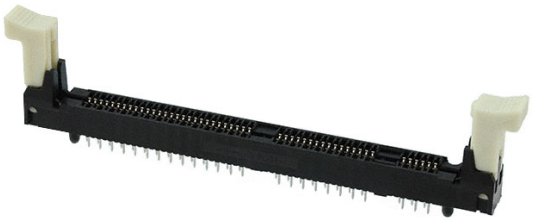

Fig 2.30: DIM100 connector.

Fig 2.31: Quadrotor platform frame.

The cross frame was purchased at a radio electronics hobby store. It is made

of carbon fiber, see *Fig* 2.31, which ensures a rigid construction and a relatively small mass. The construction weighs 240 g.

Chapter 3
Inertial navigation

As discussed in the introduction the task was to find suitable sensors to perform measurement of a vehicles accelerations vector in order to calculate the current position, primarily in the horizontal plane. The lateral movement will be limited to one axis at a time. As before, the sensors had to be readily available, reasonable in price and suitable for the application on board a quadrotor with a movement pattern containing accelerations up to 2 m/s^2 and speeds of up to 8 m/s.

The movement in the horizontal plane, s_h, is calculated as the integral of the integral of the acceleration with respect to time, i.e

$$v_h = \int a_h dt \tag{3.1}$$

$$s_h = \int v_h dt \tag{3.2}$$

where a_h is acceleration and v_h is velocity in the horizontal plane.

Since the accelerometer is fixed to the vehicle, one must compensate for the sensors possible angular deviation from the horizontal plane, described by the angle α, in conjunction with adjustments for acceleration readings contributed by the gravity, which also is related to α. With this in mind and by the means if a simple vector analysis the measured acceleration, a_m, can be described as in equation (3.3).

$$a_m = a_h \cdot \cos(\alpha) + a_g \tag{3.3}$$

where $a_g = \sin(\alpha) \cdot$ g, a_h denotes the horizontal acceleration, g is equal to 9.81 m/s^2 and the angle α is given by the strapped down sensor. Do note that (3.3) is for the simplified case when the craft is fixed in such a manner that no movement in the perpendicular directions are allowed.

The acceleration, a_h, can be approximated by the equations (3.4) and (3.5) when the lateral movement of the craft starts from the equilibrium state.

$$F_h = \sin(\alpha) \cdot F_v \tag{3.4}$$

$$a_h = \frac{F_h}{m} \text{ m/s}^2 \tag{3.5}$$

Where F_v is the vertical thrust, F_h and a_h is the horizontal force and acceleration. Do note $F_v = m \cdot g$ while the craft is in its equilibrium state.

As an example, if the quadrotor is in the equilibrium state and the lateral movement will be conducted through moderate pitch or roll movements by $\alpha = 10^\circ$ instantaneous change of the attitude with a duration of 2 s, an acceleration of 1.7 m/s^2 will be obtained. This can be calculated by using the equations above, if one allows to neglect the change in F_v during the change of the angular inclination, which will be insignificant for small α.

An acceleration of 1.7 m/s^2 for 2 s yields a velocity of 3.4 m/s. This speed is to great to start performing the navigation with (for safety reasons). A more suitable case would be if one commands the platform to have a angular inclination of $\alpha = 3^\circ$ it will accelerate with 0.5 m/s^2, if the acceleration has a duration of 2 s the velocity will be 1 m/s^2. After 100 s a horizontal movement of 100 m will have taken place. A good result would be to achieve an accuracy of 1 m or in other words 1% error.

If the gyroscope signal is high pass filtered with an appropriate cut-off frequency, so that attenuation of the 2 s signal is made negligible while the typical slow bias drift, see the previous description regarding this subject, is suppressed, the attitude for the speed control can be measured and controlled with the help of the gyroscope sensor.

For the purpose the authors selected the previously used accelerometer, namely the ADXL203 ($\pm$ 1.7 g) and a gyroscope type LY330ALH (300 $^\circ$/s).

3.1 Estimations of the measurement accuracy

With the 12-bit analog-to-digital converter of the Piccolo MCU and a span of 3.3 V, the value of 1 qu is 0.8 mV. By assuming that the maximum error is 1 qu, the error of the acceleration becomes as shown in (3.6)

$$0.8 \cdot \text{mV} \cdot 9.81 \cdot \frac{\text{m/s}^2}{\text{V}} \approx 8 \text{ mm/s}^2 \tag{3.6}$$

(The sensitivity of the ADXL203CE is 1 V/g).

If the desired acceleration is 0.5 m/s^2 and the actual acceleration is 0.508 m/s^2

for a duration of 2 s, the distance travelled during 100 s then becomes 101.6 m instead of 100 meters. This in accordance with the equations below.

$$v = a \cdot t$$
$$s = v \cdot t$$

A control algorithm is meant to keep the desired velocity during the flight with the help of the described pulses of attitude change. Potential errors are neglected for now because of their stochastic nature.

Another source of error is the drift of the offset (denoted by o_e) in the AD-converter, which is presumed to be in the range of 2-20 ppm and is described by equation (3.7)

$$V_{err} = o_e \cdot V_{ref} \approx 6.6 \text{ to } 66\text{uV} \tag{3.7}$$

The expected error corresponds to 65 to 650 um/s^2 which yields an error of 0.3 to 3 meters during 100 s.

If the output from the accelerometer is low pass filtered with a cut-off frequency of 100 Hz the smallest theoretically detectable acceleration can be calculated by the equation (3.8) which is provided by the manufacturer.

$$a_{min} = 110 \cdot \frac{\text{ug}}{\sqrt{\text{Hz}}} \approx 11\text{mm/s}^2 \tag{3.8}$$

The result of equation (3.8) is on par with the bit error mentioned above.

If one provides a low pass filter, f_c = 100 Hz, for the gyroscope the least detectable angular velocity will be 0.14°/s. This is calculated with the manufacturer provided data, see equation (3.9) below.

$$w_{min} = \frac{0.014^\circ/\text{s}}{\sqrt{(\text{Hz}}} \tag{3.9}$$

The authors chose a sampling frequency of 1 kHz. This because the fact that the acceleration readings has to be described in relatively high detail in order to approximate the position accurately.

3.2 Computational cost

As described above, the horizontal acceleration, a_h, measured whilst the quadrotor is accelerating horizontally is described by (3.10)

$$a_h = \frac{a_m - a_g}{\cos(\alpha)} \tag{3.10}$$

where

$$a_g = \sin(\alpha) \cdot 9.81\text{m/s}^2 \tag{3.11}$$

The evaluation of $\sin(\alpha)$ and $\cos(\alpha)$ often comes with a high cost for a processing unit (without dedicated hardware support). The beloved Taylor expansions for $\sin(\alpha)$ and $\cos(\alpha)$ can be utilized to approximate the given trigonometric functions. The number of terms is chosen by considering the required accuracy. With three terms included the error is guaranteed to be less than $3{\cdot}10^{-6}$ for $\cos(\alpha)$ and $1.4{\cdot}10^{-6}$ for $\sin(\alpha)$ while $|\alpha| \leq 20^\circ$. For this level of accuracy 6 multiplications and 2 summations are needed to approximate $\sin(\alpha)$ if it is calculated as showed in the pseudo code below.

```
x2  = x*x;
sin = x*(a0 - x2*(a3 + x2*a5));
//a0, a3 and a5 is previously calculated
```

This since the Taylor expansion of $\sin(\alpha)$ is as showed in (3.12) beneath.

$$\sin(x) \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} \tag{3.12}$$

In a similar manner it can be determined that $\cos(\alpha)$ will need 4 multiplications and 2 subtractions, see the pseudo code below.

```
x2 = x*x;
cos = 1 - x2*(a2 + x2*a4));
//a2 and a4 is previously calculated
```

The number of multiplications thus far are 9 (since it is only necessary to compute x^2 once) and the required amount of summations are 4. After performing three additional operations, namely one multiplication, one division and one summation a_h is obtained.

As mentioned before, a numeric quadrature has to be performed twice to determine the the position of the craft. This will add 4 multiplications and 2 summations. Note that the division by two is replaced with a multiplication of 0.5 because of the fact that divisions has a higher cost. Hence 14 multiplications, 7 summations and 1 division is necessary. All of these computations are for one

axis and the purposed navigation will take place in the horizontal plane which adds another axis. The resulting number of computations is thus equal to 28 multiplications, 14 summations and 2 divisions.

3.3 Estimation of execution time

It is important to understand how long the execution time will be for the CLA tasks so that a task call will not be missed.
Since there is a division present in INS computations, one has to use the provided instruction that gives a reciprocal approximation of x. As mentioned before there is a necessity to perform an iterative algorithm to achieve full single precision. This algorithm has to be examined in order to determine the execution time. This is done with the code snippet below.

```
1  MEINVF32   MR1, MR2          ; MR1 = Reciprocal approx. of 1/x
2  MMPYF32    MR3, MR1, MR2     ; MR1 = (1/x) * x
3  MSUBF32    MR3, 2.0, MR3     ; MR3 = 2.0 - (1/x) * x
4  MMPYF32    MR1, MR1, MR3     ; MR1 = (1/x)*(2.0 - 1/x * x)
5  MMPYF32    MR3, MR1, MR2     ; MR1 = (1/x) * x
6  MSUBF32    MR3, 2.0, MR3     ; MR3 = 2.0 - (1/x) * x
7  MMPYF32    MR1, MR1, MR3     ; MR1 = (1/x)*(2.0 - 1/x * x)
```

Listing 3.1: Clock cycle count for the inverse of x.

The above implies that it will take 7 clock cycle to perform the the inverse of x and hence 8 cycles to compute a division y/x (memory reads are not included). With this in mind it it possible to determine the execution time of the INS algorithm. See *Tab* 3.1.

Tab 3.1: Summation of execution time for the INS algorithm.

Operation	Count	Cycle/Op	Total Time
Multiplication	28	1	311.08 ns
Division	2	8	177.76 ns
Summation	14	1	155.54 ns
Sum		58	644.44 ns

Chapter 4

Results and performance

In this chapter the measurement results of the sensor filters, the mechanical construction and performance of the quadrotor platform will be addressed. The measurements regarding inertial navigation will also be displayed and discussed. Furthermore possible improvements will also be suggested along with a presentation of future work.

4.1 Quadrotor Platform

Sensor - Pitch and roll

The sensor filter was designed with classical operational amplifiers and discrete components, the schematic can be found in *Fig* A.2 in the appendix. Of course there had to be made some modifications to the simplified version seen in *Fig* 2.16 that was presented earlier. The low pass filter in the path of the accelerometer was made so that it easily can be configured to a second, third, fourth or fifth order filter (to better suppress acceleration readings originating from lateral movements with frequencies higher than f_c). The DC bias from the accelerometer had to be eliminated, this was accomplished by a summator. The summator was equipped with a variable resistor in the feedback loop which made it possible to match the two sensors amplitude.

There were also changes in the path for the gyroscope. The integrator has a resistor in parallel with the feedback capacitor (τ relatively large). This to assure that the operational amplifier will not be saturated. Furthermore a first order high pass filter (τ relatively small) was also put after the integrator to compensate for it's internal offset (which also gets integrated).

In *Fig* 4.1 one can see the output from the low pass filtered accelerometer and the integral of the output from the gyroscope and the resulting signal from the sensor filter.

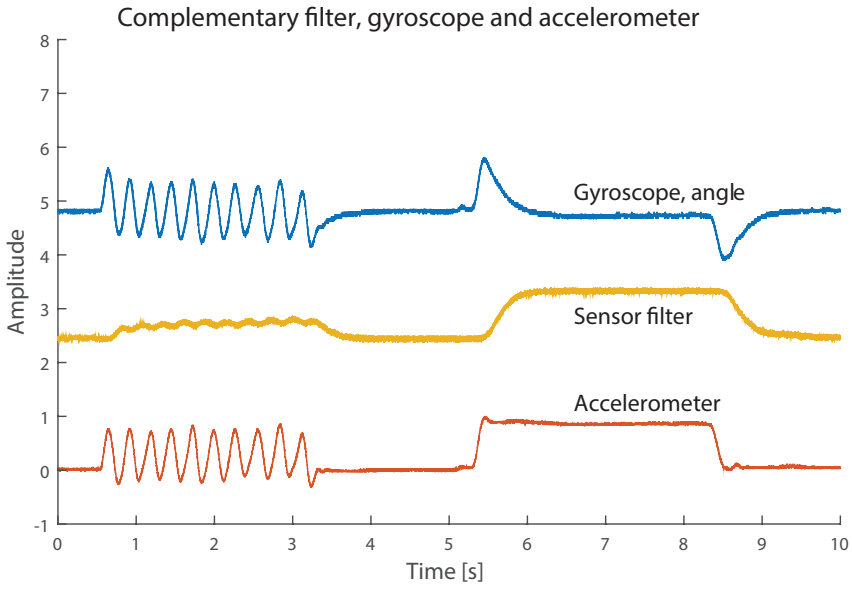

Fig 4.1: Sensor filter plot

As seen the response time is significantly improved compared to the accelerometer signal and it shows no drift tendency at all.

Due to the design of the filter, on start up, when the power is switched on, the signal moves asymptotically to the stationary value. See *Fig* 4.2.
The result is the combined step response from both the gyroscope and accelerometer path. As seen it takes approximately 100 s before the signal reaches the "zero value". Thus initialization sequence of the control algorithm has to be done after the signal stationary value.

In *Fig* 4.3, the PCB module is showed. Two sensor filters, one on the front and one on the back, are mounted on a larger PCB which will be inserted onto the motherboard.

A couple of extra features were added on the larger PCB. The first addition was an amplification, since the craft is restricted to only change the angles ϕ and θ to $\pm$ 20° it is convenient if this movement utilizes the entire span of the ADC. The need for the second addition arises whilst introducing the first. The maximum output from the sensor filter can now exceed the limits of the AD-converter and thus the ADC must be protected from this. And lastly, since the signal bias from

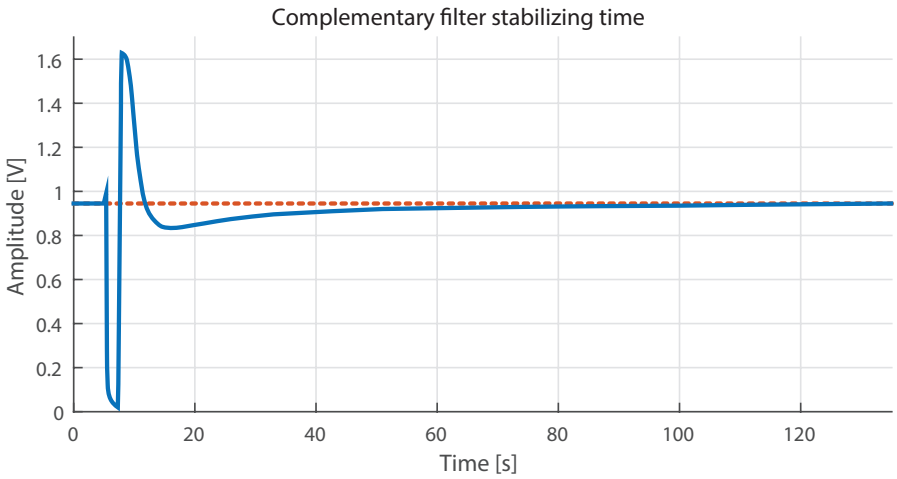

Fig 4.2: Sensor filter stabilizing time.

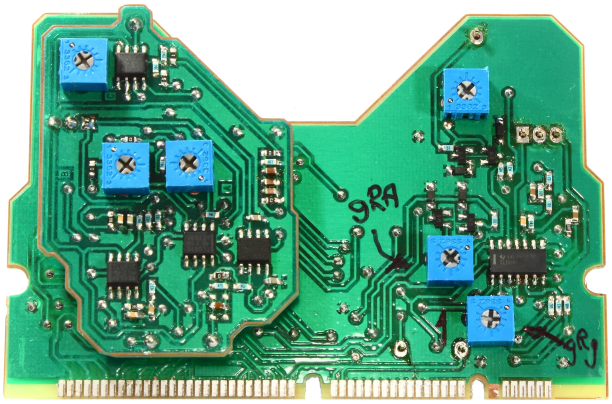

Fig 4.3: Two sensor filters (one on back) positioned on PCB which will be inserted on to the motherboard

the sensor filter is 0 V the bias has to be adjusted so that the signal is placed in the center of the ADC span, this was previously done on the motherboard and is for that reason missing in the sensor filter schematic.

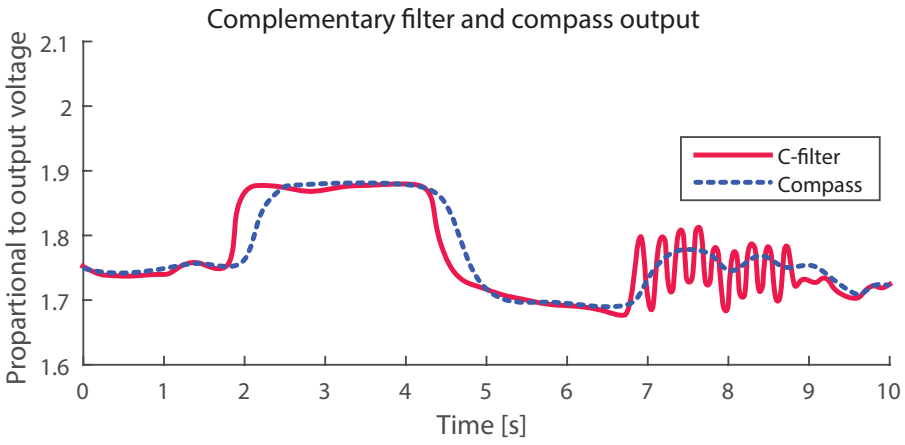

Fig 4.4: Comparison between the compass and sensor filter output.

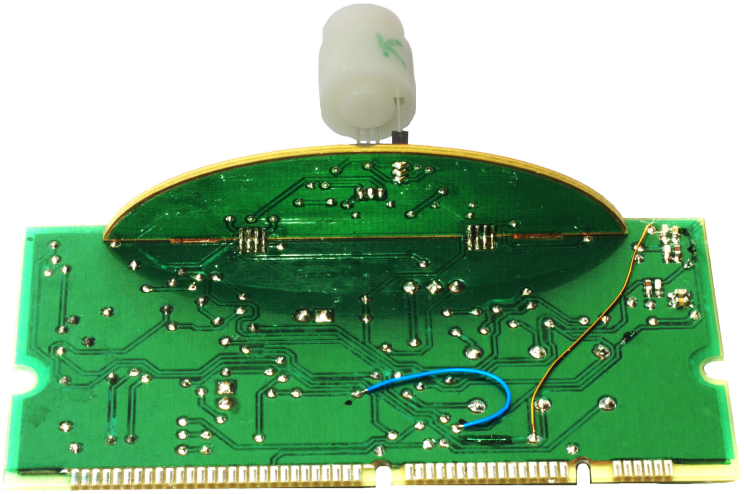

Fig 4.5: The front of the compass module.

Sensors - Yaw axis

The compass was realized with analog electronics and could relatively easily be tuned to an estimated accuracy of less than one percent of error which corresponds to $\pm 3^{\circ}$. The resolution of the compass is 6.4 mV/$^{\circ}$. For the schematic, please refer to *Fig* A.1 in the appendix.

As for the sensor filter made for the angles ϕ and θ the improvement in the re-

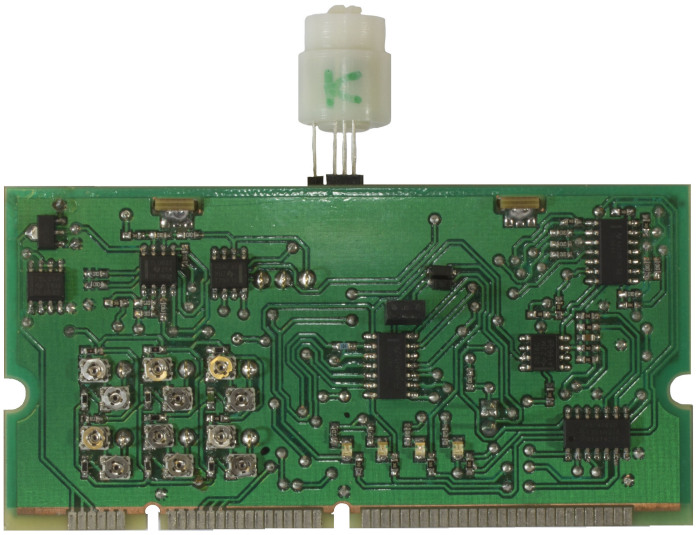

Fig 4.6: The back of the compass module.

sponse time is significant and is visualized in *Fig* 4.4. The combined signal from the gyroscope and the compass is at least 300 ms faster than the signal from the compass. In *Fig* 4.5 and 4.6, one can see the actual compass.

Power distribution

A topic that was not mentioned previously is the quadrotor power distribution. The average battery voltage during a charge cycle of a 4 cell is 14.8 V (3.7 V / cell). This voltage has to be converted to $\pm$ 15 *V*, +12 V, $\pm$ 5 *V* and +3.3 *V* as basic supplying voltages for the electronics. This was done in two steps, first via two opto isolated DC/DC converters, one with the output voltage of $\pm$ 15 V and one with +12 V. After this stage low drop-out regulators were used. For more details see *Fig* 4.7 and 4.8.

In a similar fashion as before (CNC mill combined with etching) the PCB below produced.

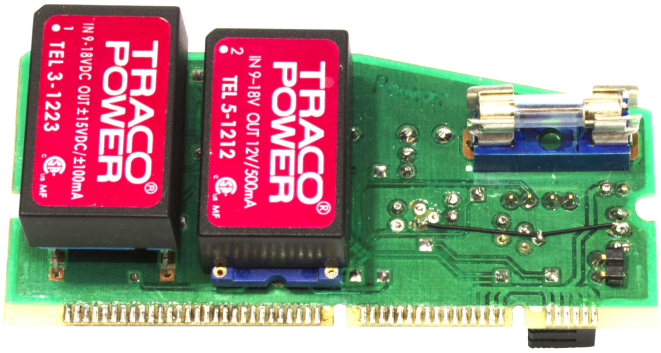

Fig 4.7: Front side of the power distribution PCB.

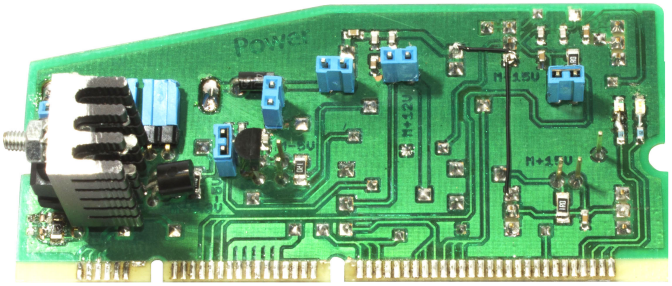

Fig 4.8: The power distribution PCB, backside.

Cross frame

As described before the frame consists of two carbon fiber tubes held together by the motherboard which was designed and created in the usual fashion. The result can be seen below. As a reminder, the top architecture of the motherboard is shown in *Fig* 4.9.

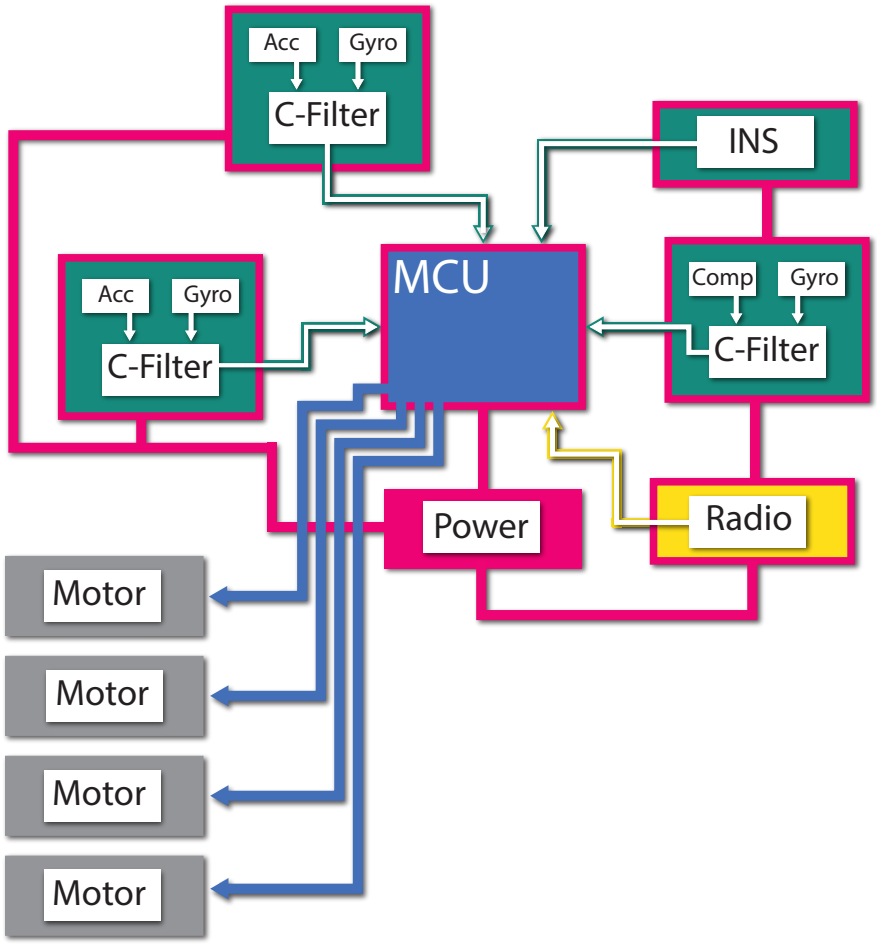

Fig 4.9: Motherboard top architecture.

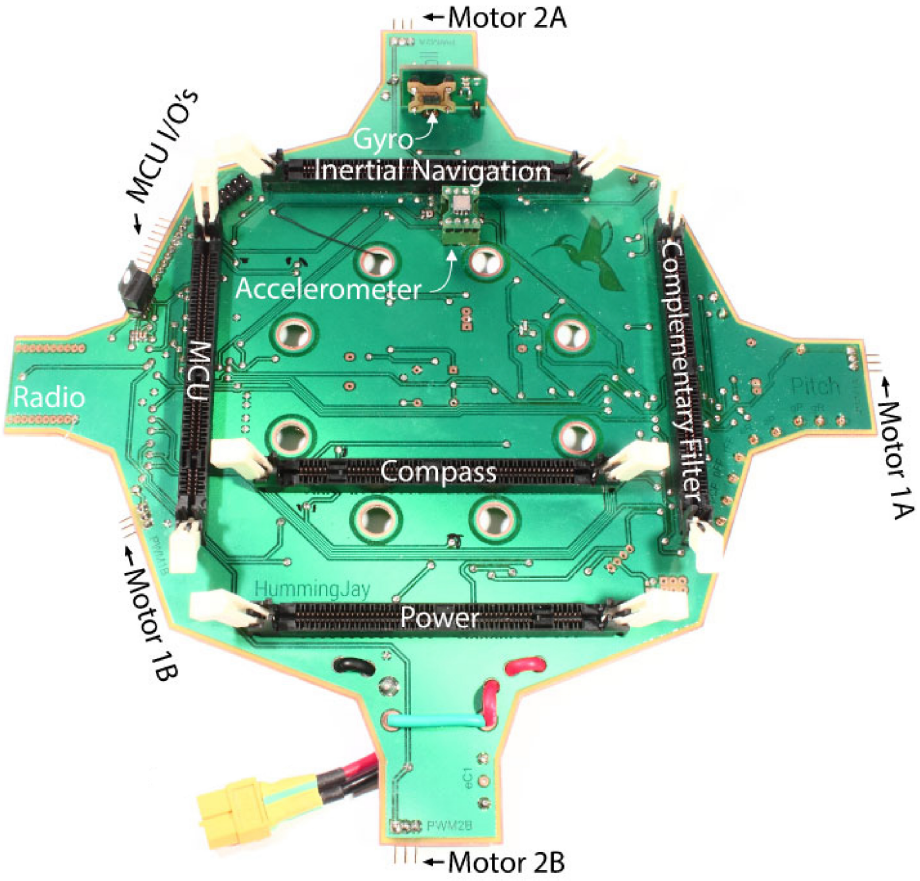

Fig 4.10: Motherboard PCB.

There were some fatal flaws in the first design of the motherboard and some modifications of the mechanical construction had to be made. The authors did not foresee the impact of the vibrations that arose from the motors and propellers. The amplitude of the vibration was severe enough to destroyed a couple of accelerometers. The author's hypothesis is that the moving conducting plate inside the sensor broke after hitting the surrounding plates repeatedly at frequencies near the sensor resonant frequency. The problem was partly solved by mounting the motors on rubber dampers see *Fig* 4.12. The achieved attenuation was not sufficient so the motherboard had to be reconstructed. It was modified so that it could be suspended in air by silicon dampers. With these precautions in conjunction with placing two smaller weights on the board the vibrations were suppressed to a satisfying level. The accelerometer was also replaced to the ADXL335 which has larger measurement span and is able to withstand higher

impact forces, 10000 g instead of 3500 g.

Fig 4.11: Motherboard suspension.

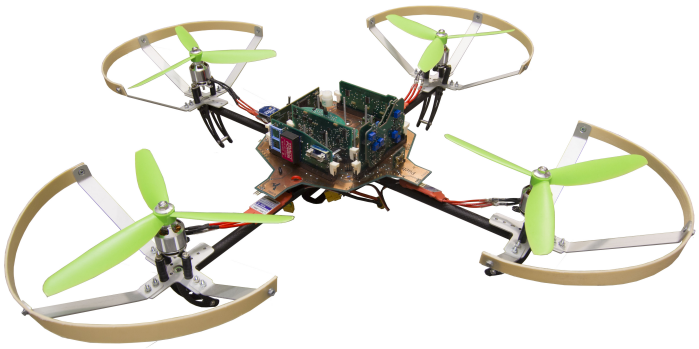

Fig 4.12: Modified frame including motor rubber suspension and motherboard silicon dampeners.

As seen above a propeller fence was created by the help of the CNC mill, aluminum plates and a plastic bucket. This also attenuated the vibrations. The completed quadrotor platform can be seen in the *Fig* 4.12.

Stability control

With the previously described control structure implemented the authors were able to reach stability in all the desired cases, yaw, pitch and roll. For a system step response please refer to *Fig* 4.13 for roll, 4.14 for pitch and 4.16 for yaw.

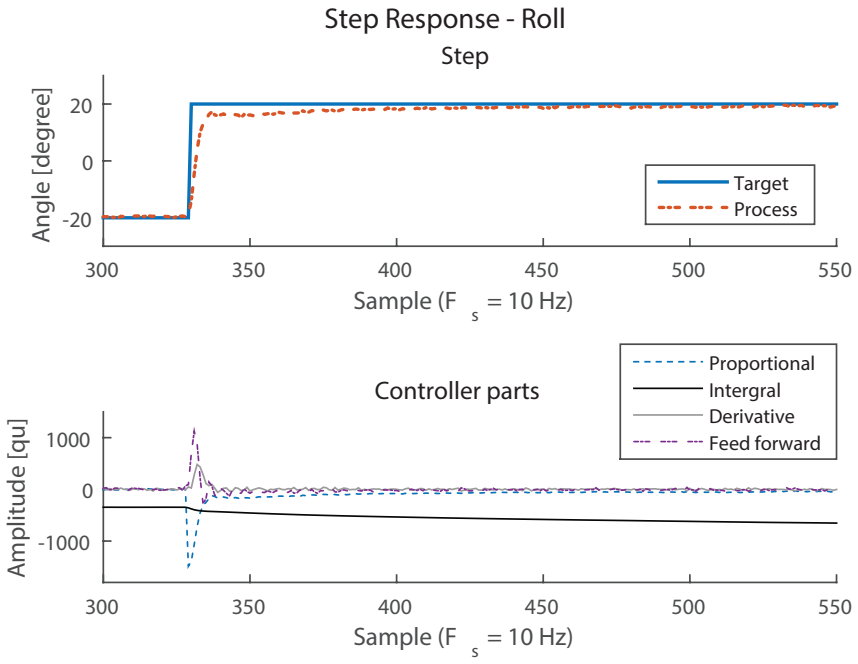

Fig 4.13: Step response, roll axis.

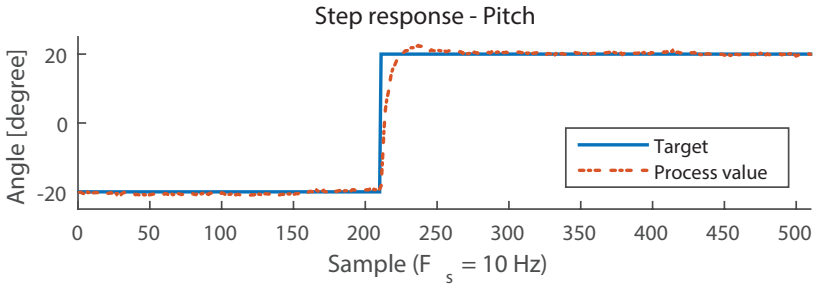

Fig 4.14: Step response, pitch axis.

The roll and pitch controllers were tested separately. The quadrotor was hanged on a string in the same test rig used for evaluating the motors. See *Fig* 4.15.

Fig 4.15: The quadrotor platform test rig.

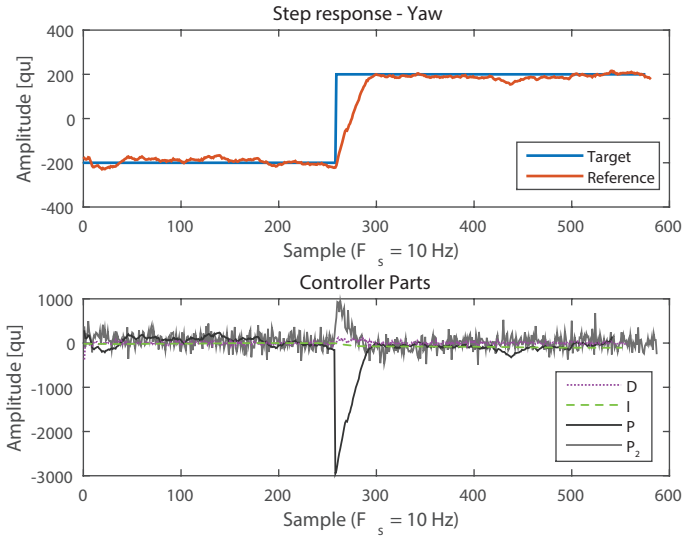

Fig 4.16: Step response, yaw axis.

The controllers were manually tuned. The main differences in the step response of the pitch and roll controllers can most likely be attributed to the sensor filters. As known, they were constructed by discrete components and thus hard to configure so that they behaved in the exact same manner.

As a curiosity, it can also be added that the implemented FIR filter made a tremendous difference regarding platform vibrations. Even though the analog signals was low pass filtered by a 2^{nd} order filter with the cut-off frequency of 20 Hz before the conversion into the digital domain. See *Fig* 4.17 and *Fig* 4.18.

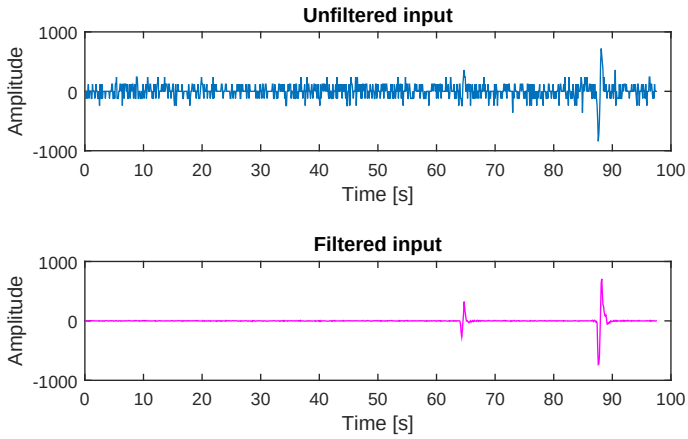

Fig 4.17: Comparison between the contribution from the derivative part when the input is filtered and unfiltered.

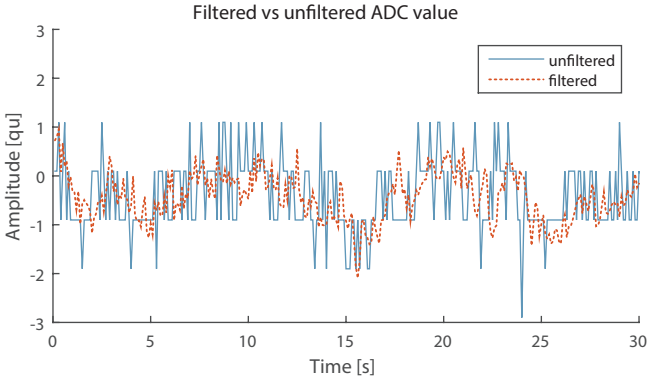

Fig 4.18: Comparison between filtered and the unfiltered input.

In *Fig* 4.19 the reader can see the platform while hovering.

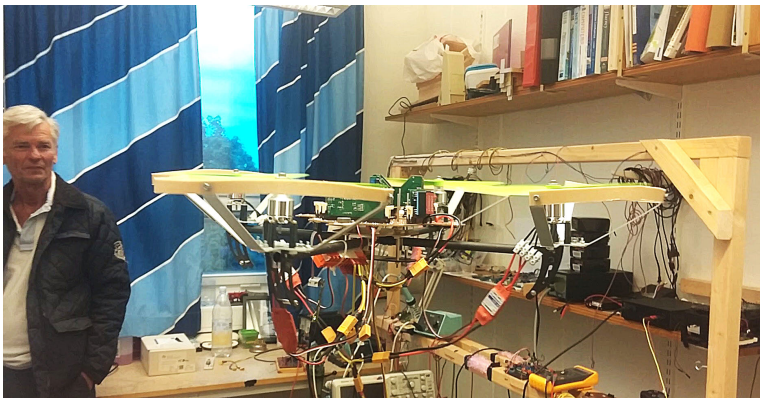

Fig 4.19: The quadrotor platform in the air.

The craft is not easily controlled by the operator though, reaching the equilibrium state is near impossible. The reasons for this will be discussed the next section.

Conclusion and possible improvements

The altitude is cumbersome to set due to the fact that the stick range goes from zero to full throttle. In other words it is to sensitive and the slightest change have great impact. If $F_v + F_g \neq 0$ (F_v denotes the force generated by the motors and F_g the force imposed by the gravitational field) the craft will either accelerate up- or downwards. Since the platform protocol only provides a 5-bit resolution of the thrust it is likely that $F_v + F_g = 0$ never occurs.

The pitch and roll sensor filters "zero value" that is subtracted from the signal before evaluating the control algorithm needs to be spot on, e.i. it has to be the value corresponding to an angular inclination of 0° and this is not always the case. If it is not the craft will drift in the horizontal plane. To eliminate the lateral movement the remote controller stick need to be set at the position so that it counteracts the drift. And as for the thrust, the stick resolution was to low. These two major flaws is what made the quadrotor hard to control.

Furthermore the motor/propeller vibrations made it hard to have sensors with higher accuracy on board, which is needed for the inertial navigation system. By building a custom made frame for the quadrotor with the vibration problem in mind it would be easier to more efficiently suppress them.

4.2 Inertial Navigation

To verify the characteristics and capabilities of a inertial navigation system based on the MEMs sensors numerous experiments and measurements were carried out, of which the following interesting results were selected to be reported.

A evaluation PCB was designed and created, with the combination of a CNC mill and etching. On the PCB four accelerometers were placed. Each signal was low pass filtered by a second order RC filter.

Measurement A - Sensor stationary drift

During the first measurement four sensors were connected to four different ADC-channels. The evaluation PCB was not moved during the measurement and had a duration of 100 s. Please see *Fig* 4.20.

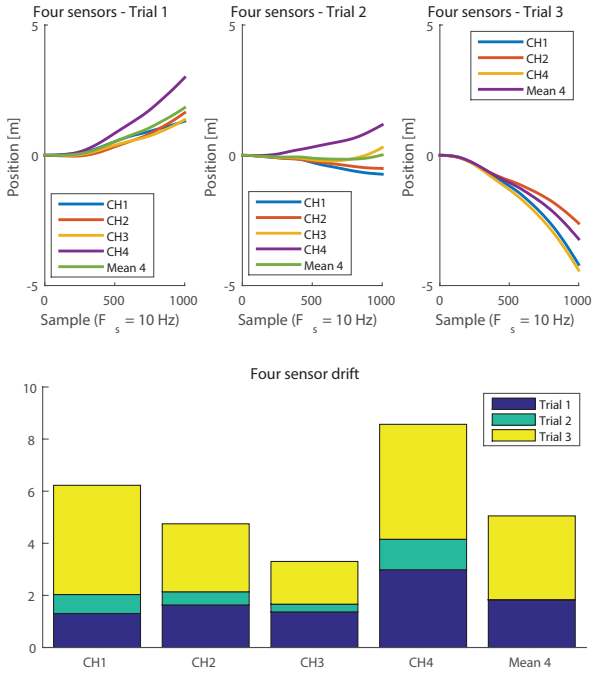

Fig 4.20: Measurment A, stationary drift.

In the graph three measurements are displayed, Trial 1, 2, and 3. For each trial the drift from each sensor and the mean of all four can be seen. Furthermore the individual absolute sum from each trial is showed as five three-part bar graphs.

Measurement B - ADC stationary drift

In order to gain more information about the ADC's drift the authors replaced the sensors in the measurement above with jumpers to GND, i.e. a guaranteed "zero value". As before, the PCB was not moved during the measurement. The obtained result can be viewed in *Fig* 4.21.

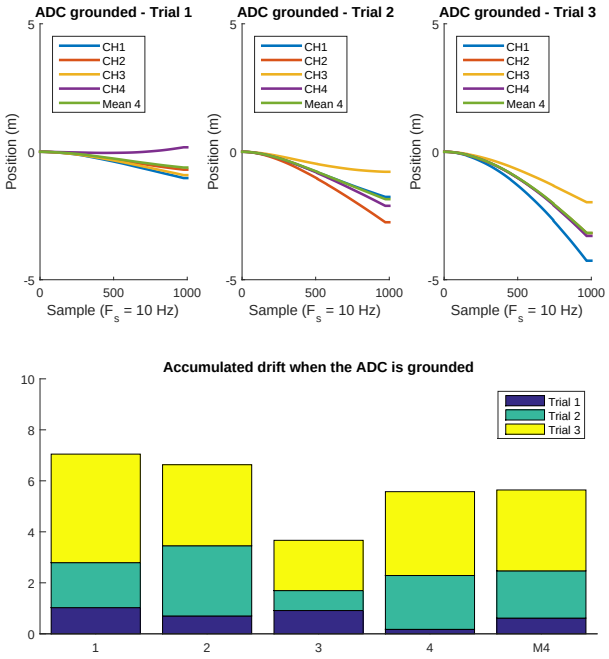

Fig 4.21: Measurment B, ADC zero valued signal drift.

Measurement C - One accelerometer to four channels

To investigate the possible differences whilst utilizing one ADXL to four ADC inputs and four ADXL to the respective inputs as per A above, an additional measurement was performed with this configuration, see *Fig* 4.22. The measurements was done without moving the PCB.

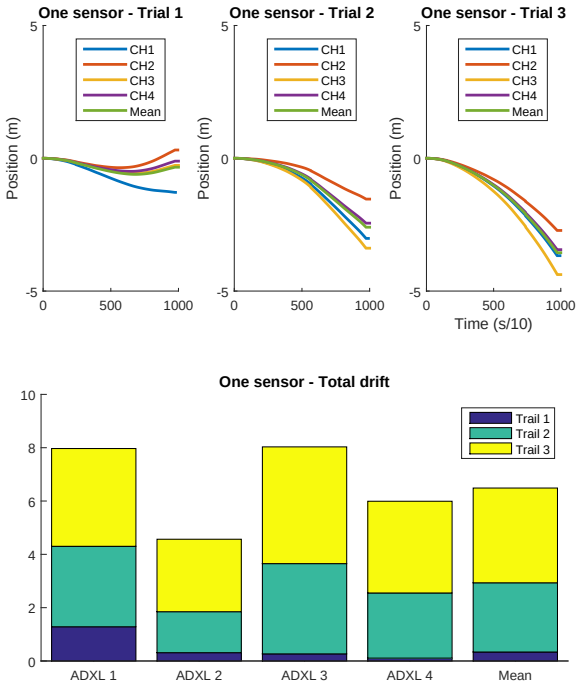

Fig 4.22: Measurment C, one sensor to four ADC channels.

Measurement D - Movement, Four accelerometers

To carry out measurements while the INS PCB was moving the authors created a controlled environment by using a motorized sled on an 1.8 m long track. See *Fig* 4.23 for a graphical reference. A motor controller based on an ATmega16 was designed and created. Three potentiometers were used to set the acceleration,

maximum speed and time for constant speed. The repeat accuracy of the motorized sled was satisfactory, the error was smaller than 0.5% between repeated runs with the same acceleration, speed and time.

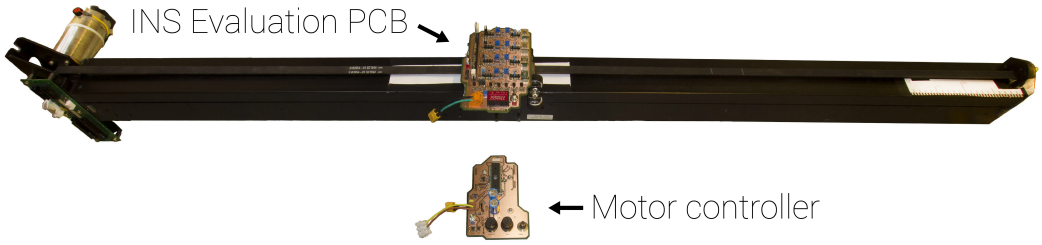

Fig 4.23: Evaluation setup - Evaluation PCB, motor controller and motorized rig.

A suitable distance and acceleration were set, where the distance was chosen to 1.30 m. A run with the measured distance of 1.29 meters was carried out, with results showed in *Fig* 4.24.

The result of the same distance traveled but with different accelerations and velocities were also conducted and showed the same results.

All of the above was repeated five times and differed no more than 1.3% between the each trial.

A new distance, 0.75 meters, with the same acceleration as above, was set via the motor controller and carried out, with measured distance of 0.76 meters. See *Fig* 4.25

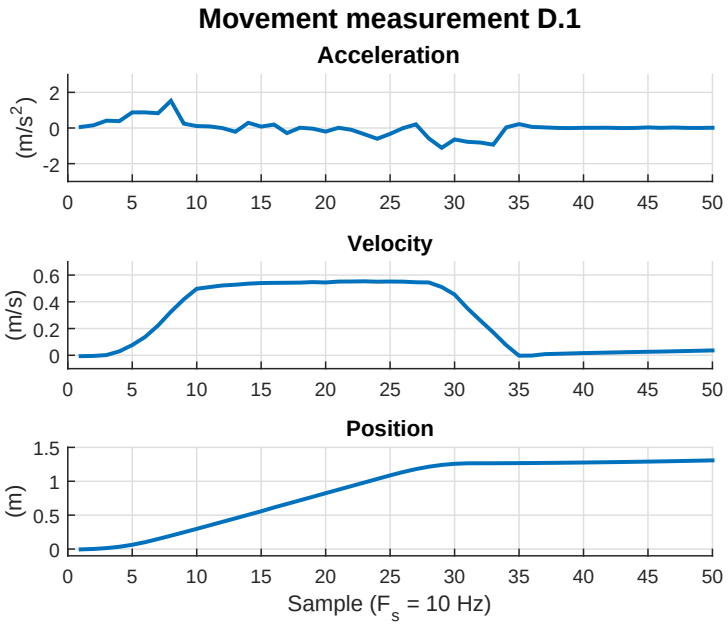

Fig 4.24: Movement measurment D.1, result 1.29 meters.

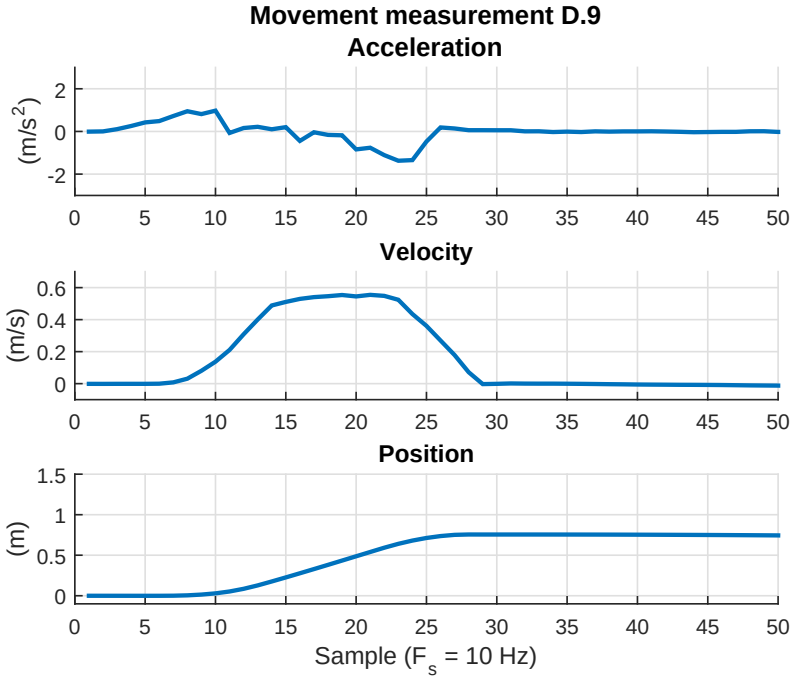

Fig 4.25: Movement measurment D.9, result 0.76 meters.

Measurement technique

The measurements were carried out with the help of the aforementioned circuit board, see appendix for schematic, and the microcontroller. Before each measurement was performed an internal ADC calibration routine was executed, subsequently, a number of dummy conversions were made and lastly, 1000 samples when the system was at rest were accumulated during 10 s. The mean of the accumulated samples was used as a zero g bias elimination constant for the continued measurements and calculations. By creating the mean out of a large number of 12-bit values which were then processed in 32-bit floating point format a high calculation precision could be achieved. The sampling frequency was 1 kHz. This may seem large but it is required to get an accurate representation of the acceleration in the digital domain. Initial measurements with 100 Hz was performed but the result had a maximum error of 30%. Higher frequencies than 1kHz did not improve the result significantly.

Conclusion and possible improvements

Measurement B shows that the drift, ranging from 0 and 3 meters per 100 s remained when the accelerometers were replaced with "zero valued signals". Because of this one can safely draw the conclusion that the drift phenomenon is mainly due to the microcontroller's AD-converter.

Measurement C shows that with respect to the drift one obtains close to the same results with one sensor versus the mean of four sensors, which confirm that the drift can be attributed to the AD-converter of the microcontroller.

Measurement D.1 and D.9 show a good capability of the system to calculate the position in a shorter time span. One can also see that the sensor signal is not always returned to the value it had before moving the sled, i.e. the sensors showed signs of a hysteresis phenomena, see *Fig* 4.26.

As a conclusion of the carried out measurements it is safe to say that the microcontrollers converter is the main limitation of the projected results due to its drift characteristics.

Furthermore the measured drift yields that a movement of 100 m can be carried out with an error no larger than a couple of percent. If the hysteresis is assumed to stay within reasonable values, it is however possible that it is evened out during a movement with varied acceleration. The results indicates that the method could in fact be used as a short term navigation technique on board a quadrotor platform with the here described restrictions.

An improvement regarding the AD-converters offset could be to have one sen-

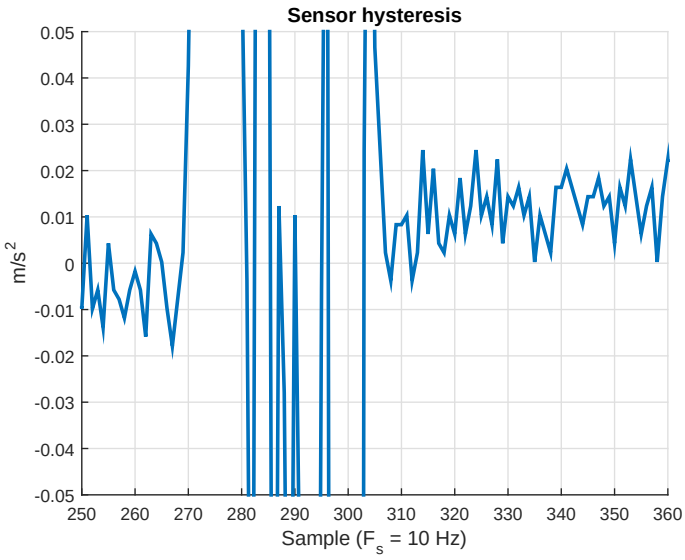

Fig 4.26: Sensor hysteresis.

sor connected to one ADC-channel while the remaining channels are connected to ground. All channels are processed equally, i.e integrated twice with respect to time. The result from the "zero valued" channels could then be used as an estimation of the drift attributed to the AD-converters offset error. The estimation can then subtracted from the position value. Furthermore it could also be of value to use better AD-converter with a smaller offset error. If the described drift is eliminated it might also be useful to use several accelerometers in order to reduce errors originating from the sensor hysteresis.

Another potential solution to the hysteresis of the accelerometer could be to subject the sensors to a constant vibration at a suitable frequency and with a relatively small amplitude. By doing this the moving conducting plate of the sensor will most likely oscillate around its "zero value" level.

4.3 Future Work

Quadrotor Platform

One of the authors has already constructed a new iteration of the quadrotor platform. This time the platform protocol is extended and allows a 10-bit resolution of the remote controller's joystick position. All of the joysticks data is packeted

in a 12 byte long flight control frame. The platform remote controller sends the flight control frame once every 50 ms but in this version the quadrotor initiates the transmission by sending one byte with the value of 0xFF. This is done to make sure that the Piccolo's main CPU, which handles the communication, is in a "low activity" state before the remote sends data. The frame consists of a two byte header, 8 byte payload and a two byte tail. See *Fig* 4.29.

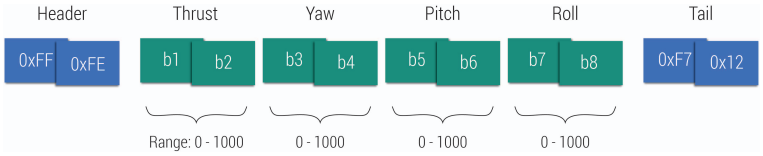

Fig 4.27: A flight control frame in the improved platform protocol.

The thrust is controlled by a stick in conjunction with a rotary encoder. The rotary encoder is used for coarse adjustment of the thrust while the stick is utilized for fine tuning the level. This is a great improvement. The control algorithm gains can be changed in flight via the new remote controller's encoder and buttons. This was an appreciated feature which made the system easier to tune. The new remote also incorporated a LCD-display where useful data is displayed such as the thrust, target angles, control algorithm gain etc. A graphical user interface (GUI) was also written which allows a computer to control the quadrotor. See *Fig* 4.29.

For the new quadrotor another sensor was used to determine the angles θ, ϕ and ψ. The sensor consists of three accelerometers and three gyroscopes. An internal processor computes the orientation quaternions (a mathematical invention to determine an objects rotation relative to a reference coordinate system) and send the data via I^2C (instructions for how to use the sensor was not available at the time the authors of the thesis started building their quadrotor, the sensor needs to be initialized properly on every start up). Three analog gyroscopes are still utilized for controlling the faster dynamic. The new motherboard now houses two serial-USB-converters which can be connected to a computer. This feature makes it possible to perform better data logging than the IDE that Texas Instrument provides. This feature also comes in handy while debugging the platform protocol as every frame can be bounced back to the computer.

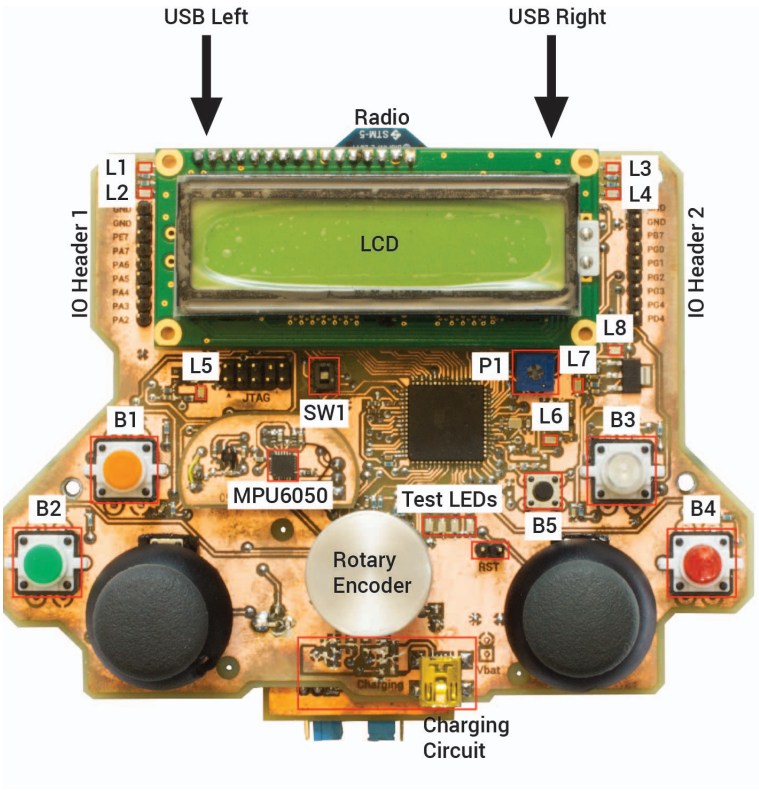

Fig 4.28: Improved remote controller.

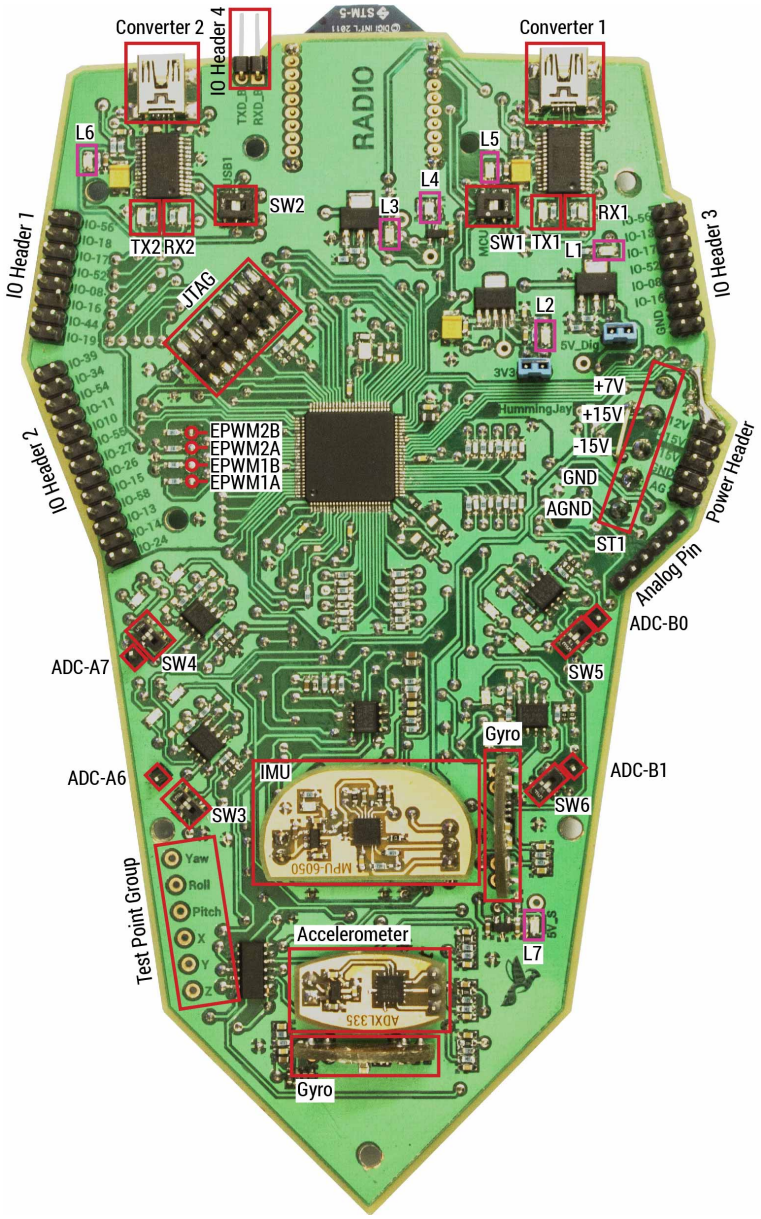

Fig 4.29: Quadrotor version two, the new motherboard.

The frame was designed and manufactured by the before mentioned author and with the intention to minimize the vibration that arises from the motors/propellers. All motors are suspended by silicone rubber strings, see *Fig*. 4.30.

The result was yet again an improvement. The vibrations were suppressed enough for allowing the craft to carrying sensors with higher precision (ADXL203).

Fig 4.30: Motor suspension system.

With the above adjustment a much more controllable quadrotor was created. See *Fig*. 4.31.

Fig 4.31: Quadrotor platform in air.

Inertial Navigation

The work done regarding the inertial navigation shows, as stated before that it has potential as a short term navigation system. One of the authors is working on implementing a velocity control algorithm for the new iteration of the quadrotor platform. The same author is also working on a new INS evaluation module where the previously described offset estimation technique and the constant vibration of the sensor is implemented.

References

[1] Torkel Glad, Lennart Ljung. *Reglerteknik Grundläggande Teori*. Studentlitteratur, Lund, 2006.

[2] John G. Proakis, Dimitris G. Manolakis. *Digital Signal Processing*. Pearson Prentice Hall, United States of America, 2007.

[3] Mats Brorsson. *Datorsystem* Studentlitteratur, Lund, 1999.

[4] Timothy Sauer. *Numerical Analysis* Pearson Addison Wesley, Boston, 2006.

[5] G. Szafranski, R. Czyba. *Different Approaches of PID Control UAV Type Quadrotor* Silesian University of Technology, Akademicka St 16, Gliwice, Poland

Appendix A

Appendix - Schematics

In the appendix the reader will find all of the schematics which was made during and presented in this thesis report. The authors strongly recommend to viewed in a digital format since the size of the schematic sheet does match the paper size of this report.

A.1 Compass

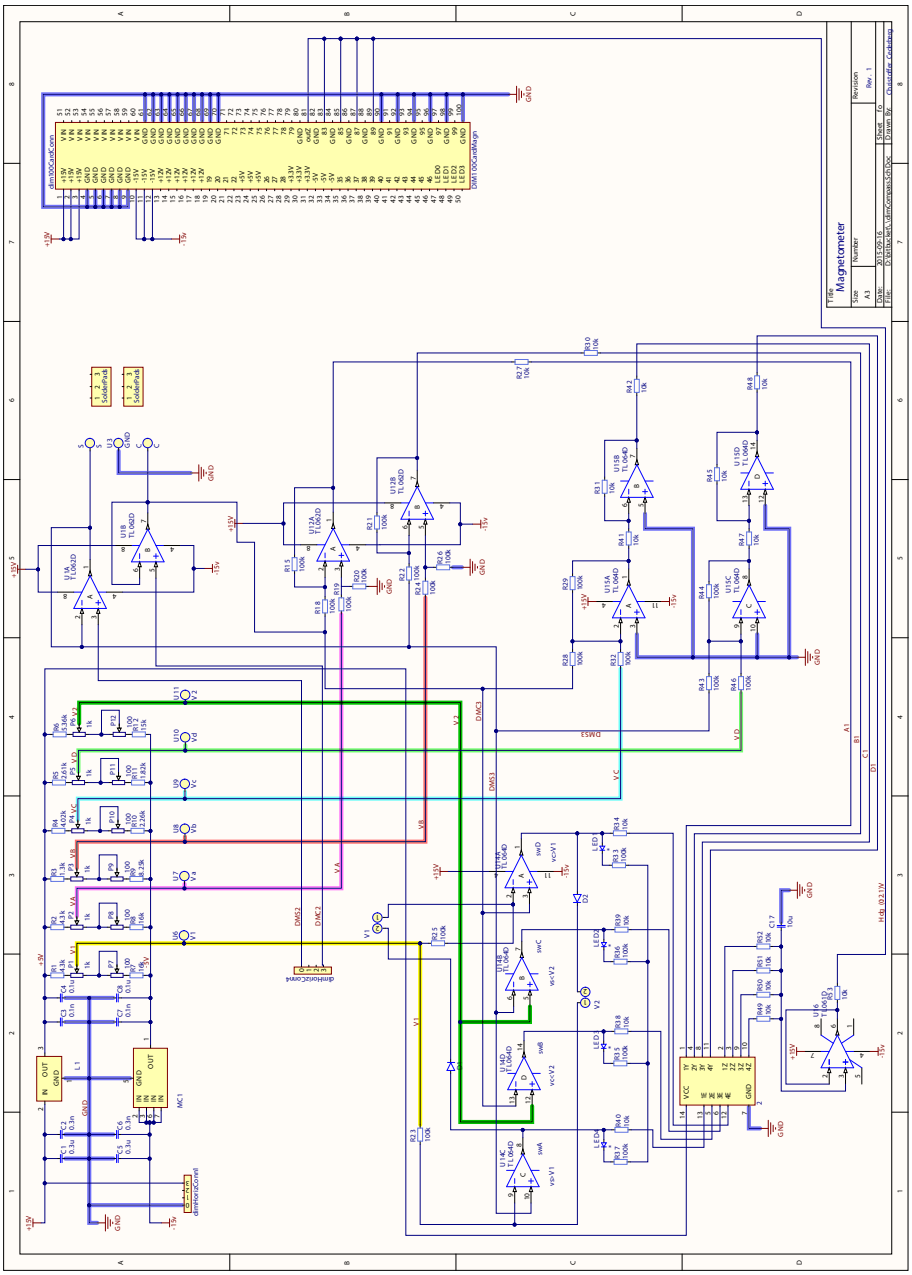

Fig A.1: Schematic of the compass.

A.2 Complemntary filter

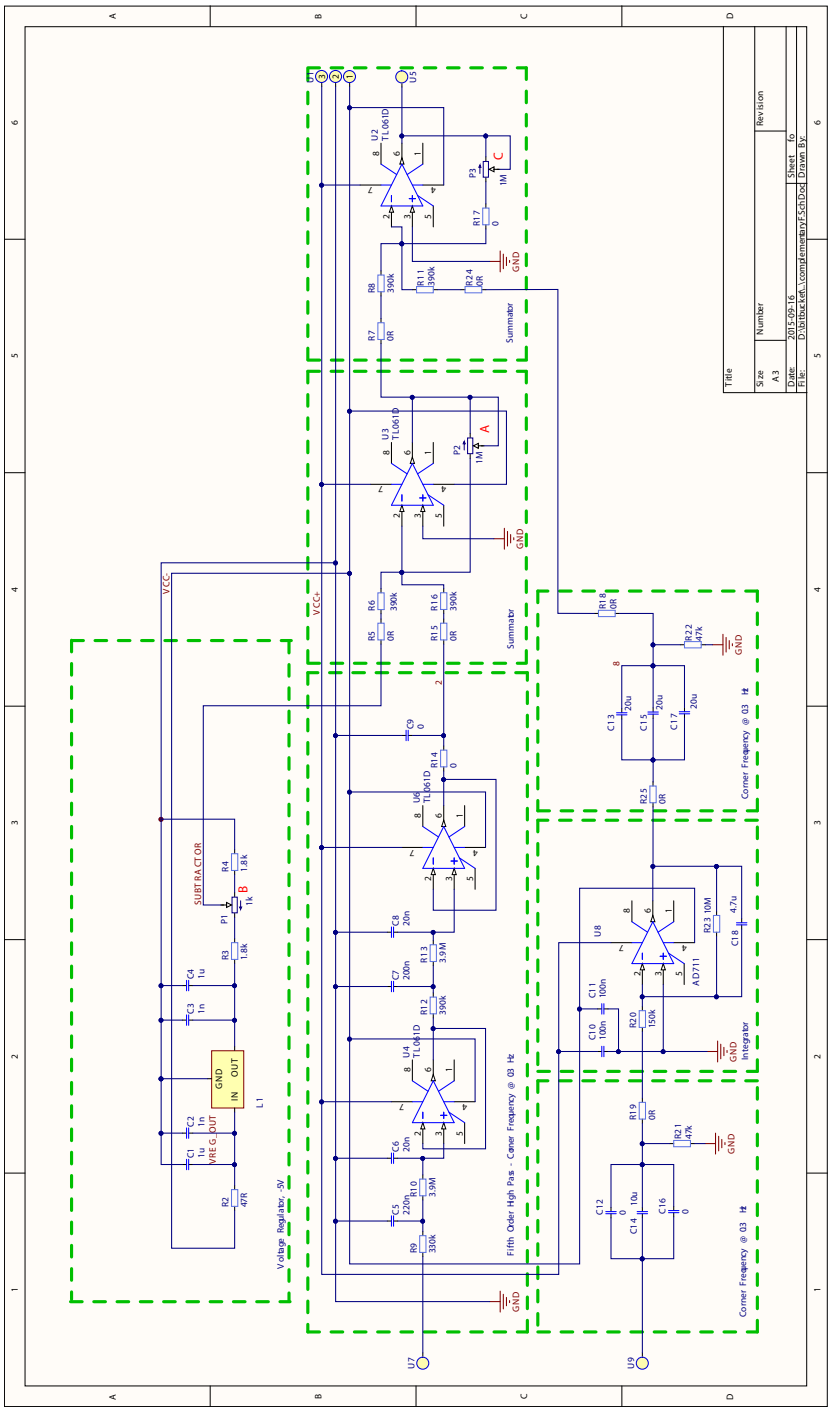

Fig A.2: The complementary filter schematic.

A.3 Motherboard

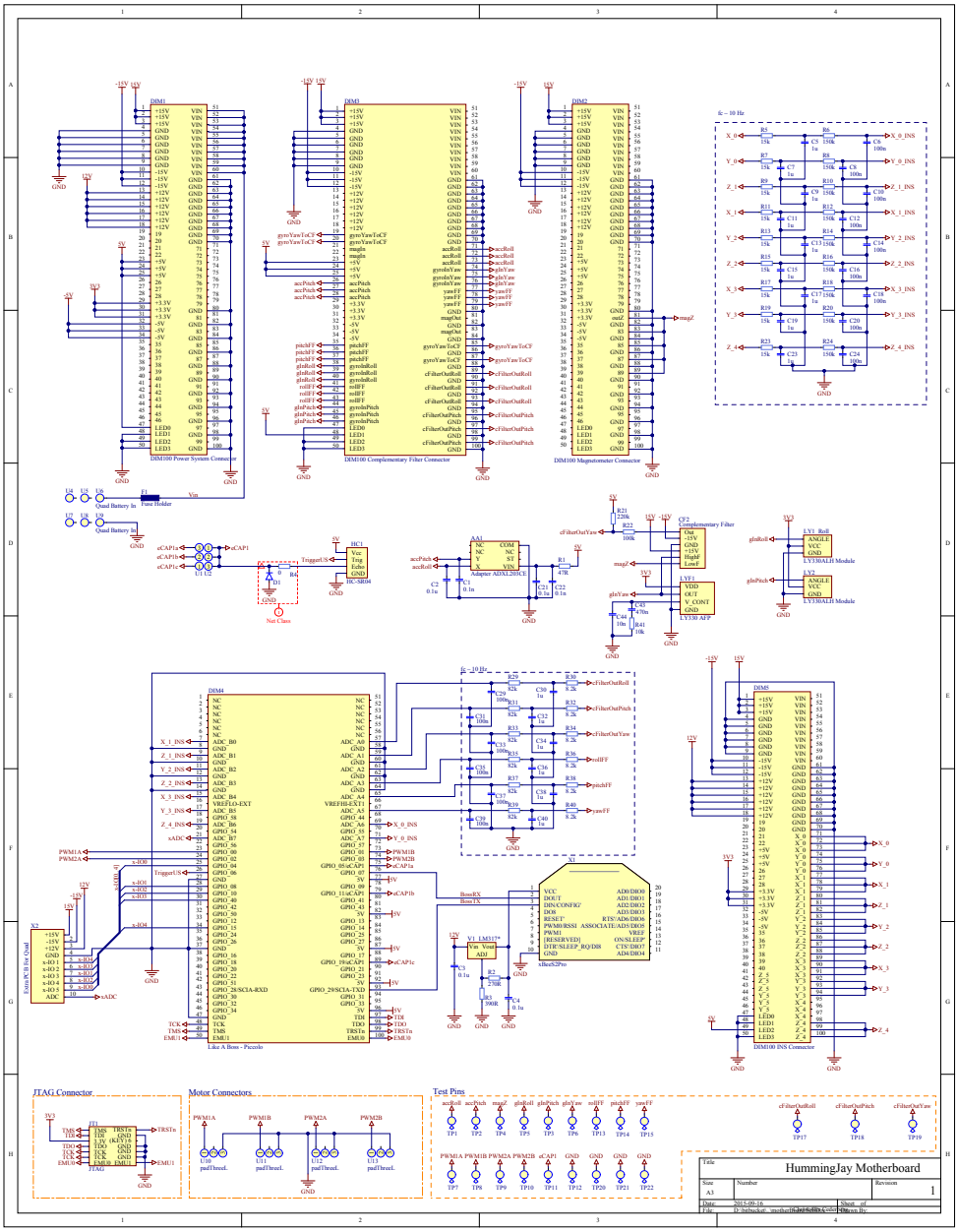

Fig A.3: Schematic of the motherBoard.

A.4 Complementary filter amplitude protection

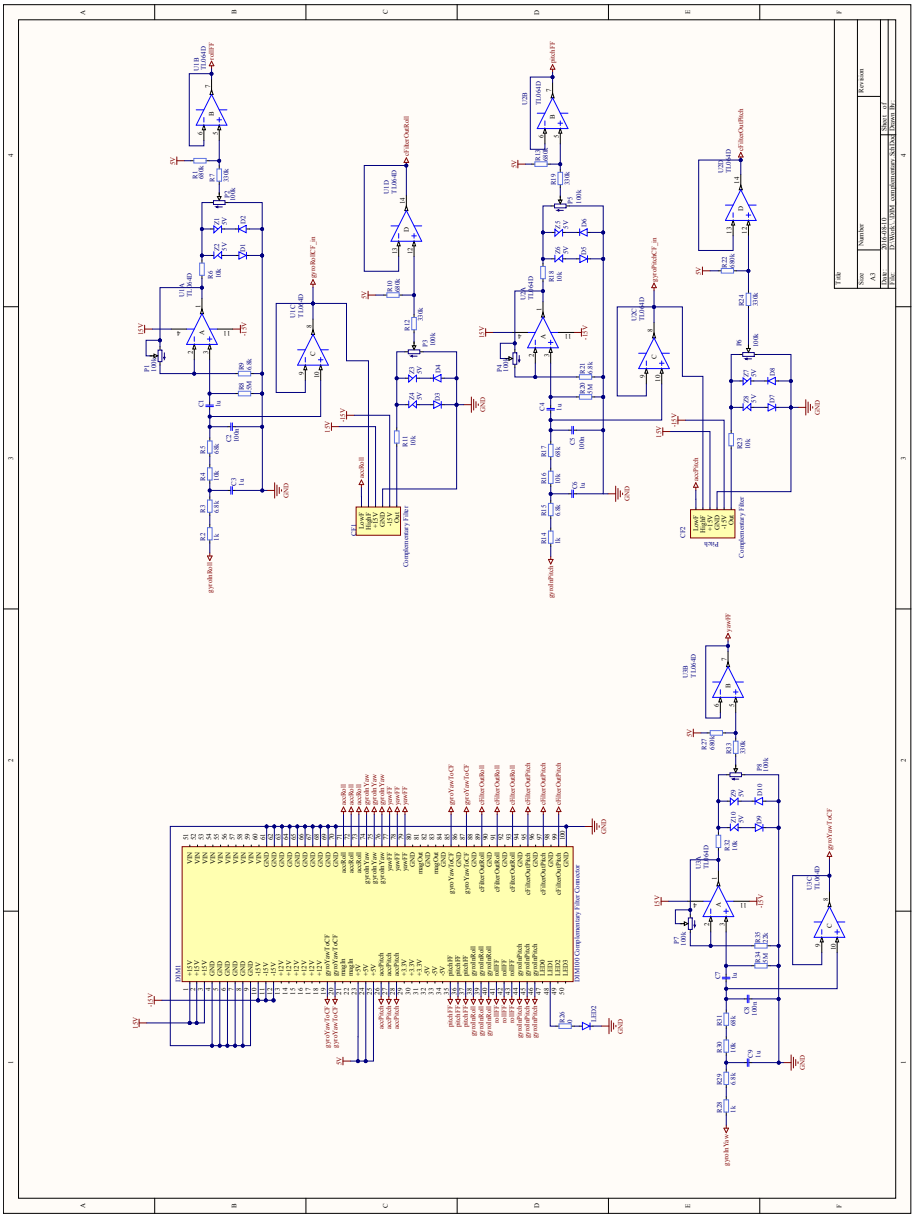

Fig A.4: Schematic of the complementary filter amplitude protection.

A.5 INS evaluation

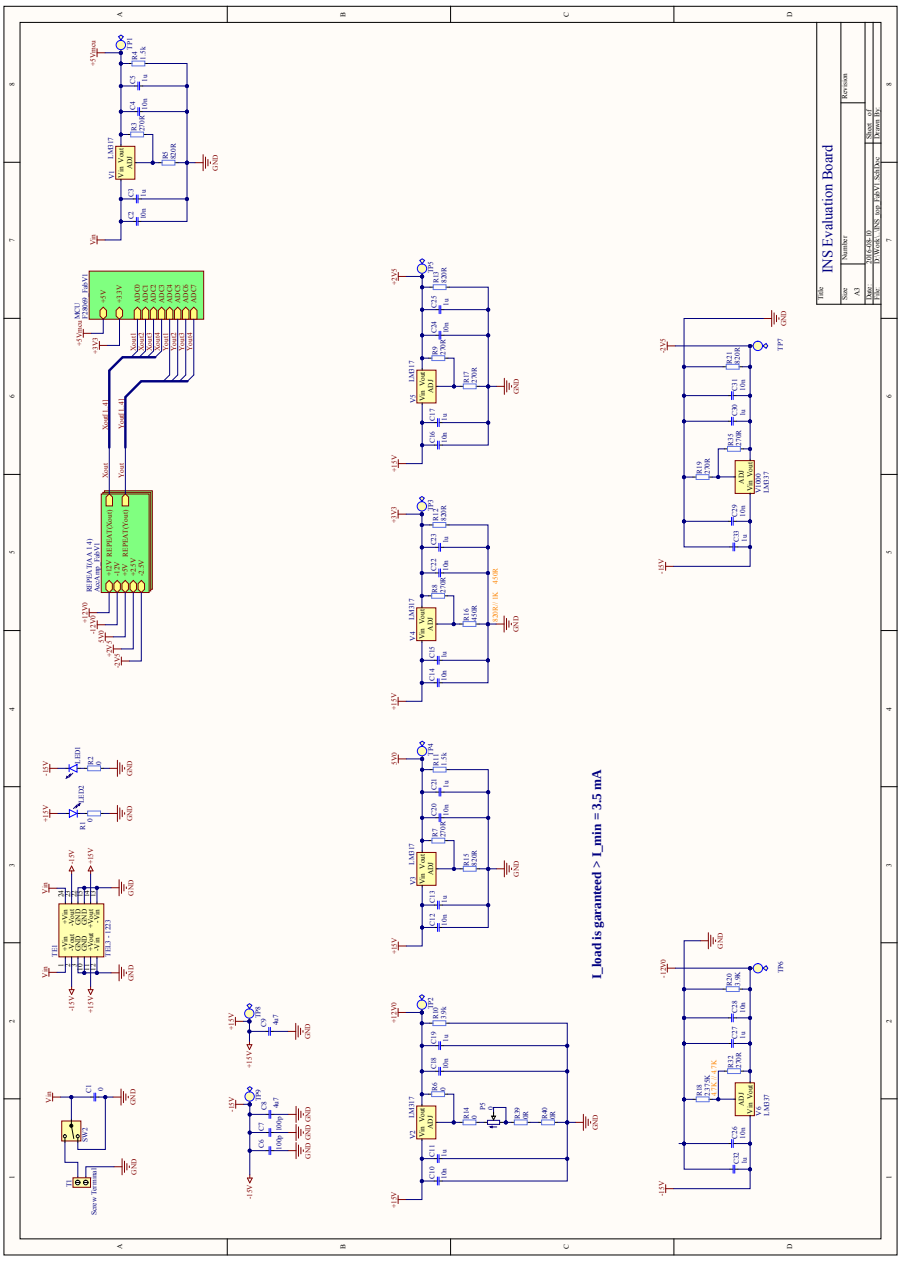

Fig A.5: Top architecture of the INS evaluation.

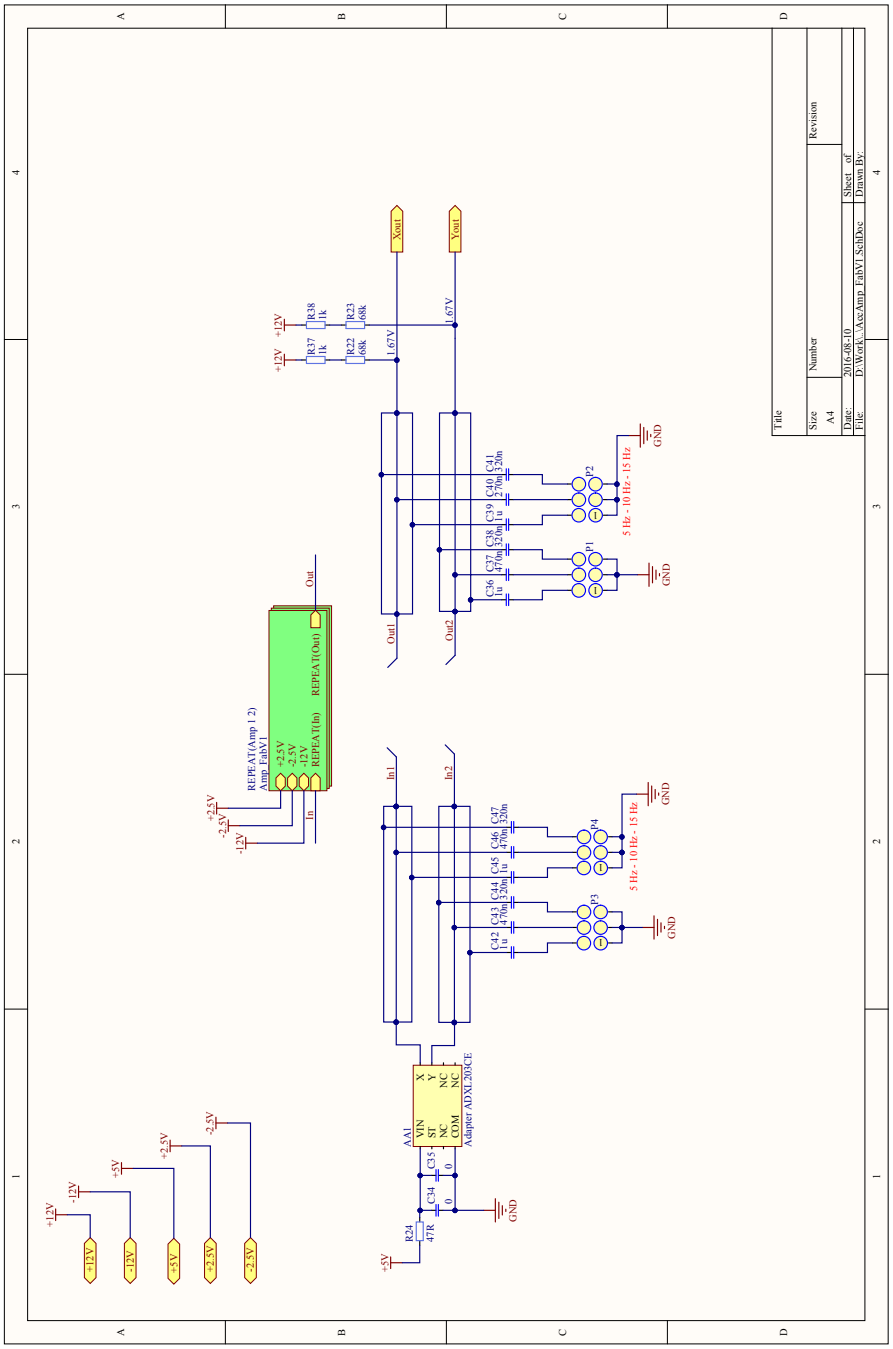

Fig A.6: The AccAmp block.

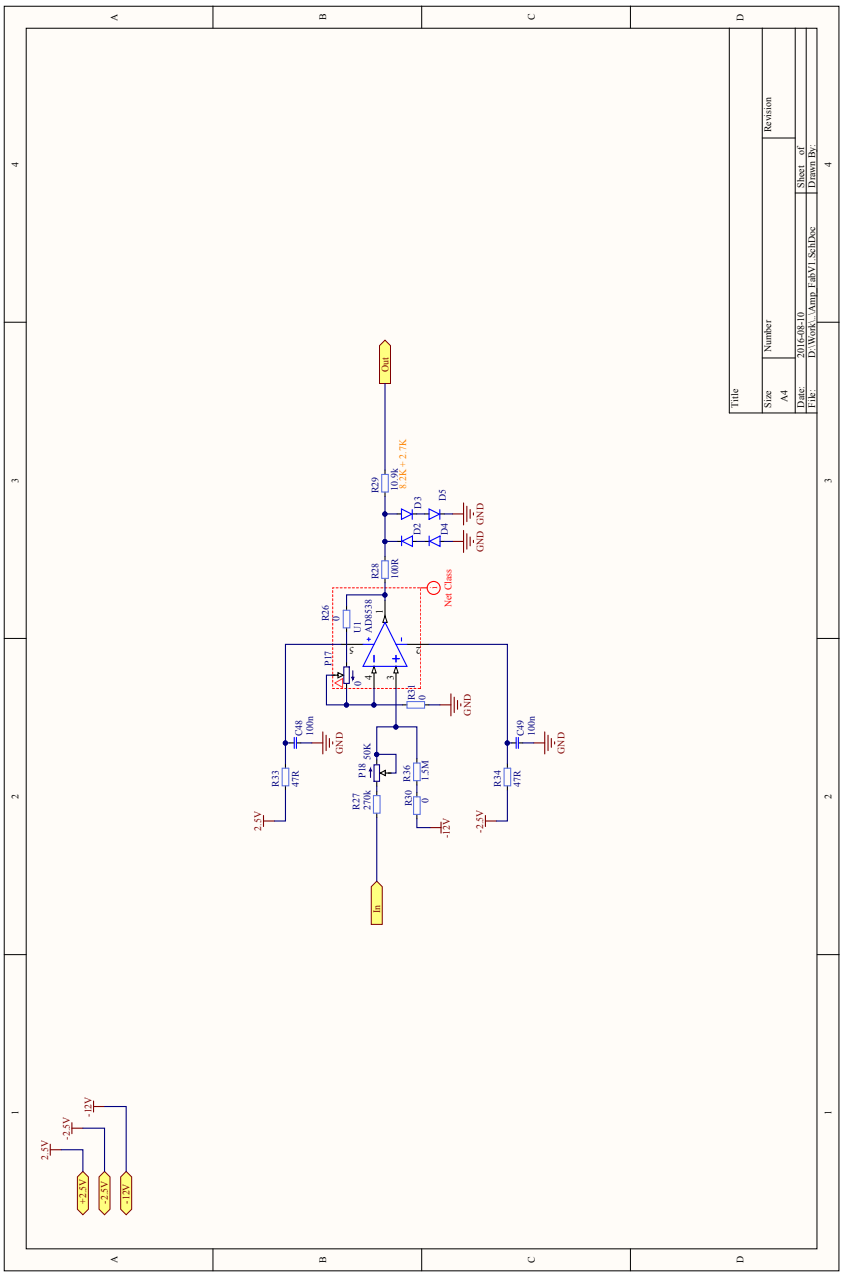

Fig A.7: Amp (Amp - amplification) block. Do note that the amplification stage was bypassed during the measurements previously documented in the report.

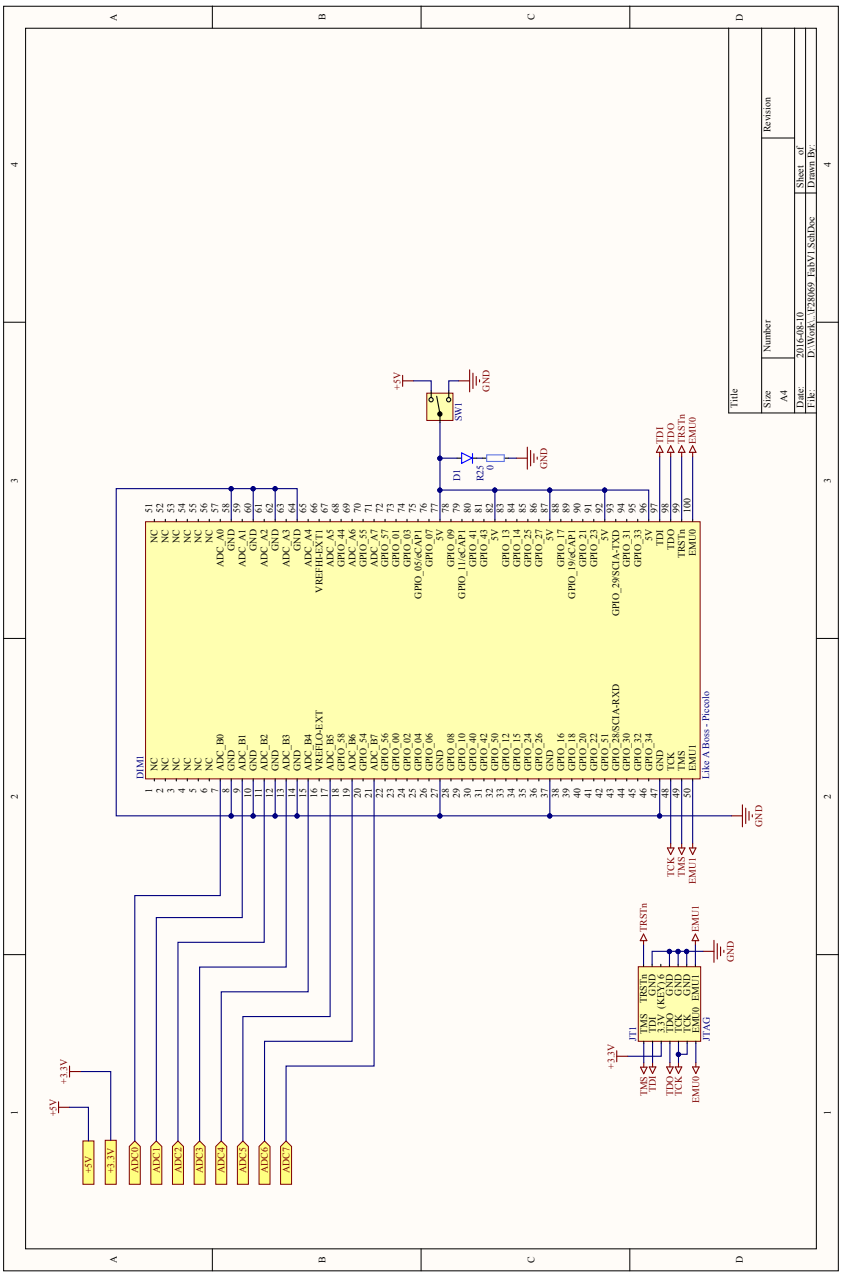

Fig A.8: MCU block.

Printed by Tryckeriet i E-huset, Lund 2017

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2017-562

http://www.eit.lth.se