



Design of a mechanical-based thrust sensing unit for multi-rotor UAVs

R. (René) Kamp

BSc Report

Committee:

Dr.ir. J.B.C. Engelen R. Hashem, MSc Prof.dr. J.C.T. Eijkel

July 2018

016RAM2018 Robotics and Mechatronics EE-Math-CS University of Twente P.O. Box 217 7500 AE Enschede The Netherlands

UNIVERSITY OF TWENTE.



BIOMEDICAL TECHNOLOGY AND TECHNICAL MEDICINE

Summary

A drone which can interact with a wall has several benefits for, for example, inspection of a surface. To get an accurate representation of the force applied to the surface a closed-loop controller on the thrust is preferred. An accurate thrust sensing unit is needed to give the amount of thrust back to the controller. A method to sense the thrust is through the use of strain gauges on the beam where the motor is attached to, but this is prone to noise generated by the motor. The assignment is to design an independent mechanical-based thrust sensing unit which will measure the thrust and also filter high frequency noise produced by the motor. It will be small, lightweight and able to serve as a gadget, which can be connected to any drone which meets the requirement of the motor. Commercial components will be discussed and analysed for the design. Lastly, the performance of the unit was validated through experiments and recommendation for a future unit were given.

Contents

1	Intr	roduction	1
	1.1	Project background	1
	1.2	Problem statement	1
	1.3	Project goals	1
	1.4	Report outline	2
2	Bac	kground	3
	2.1	Aerodynamic forces	3
	2.2	BLDC motors and ESCs	4
	2.3	Sensors and Wheatstone bridge	4
3	Mea	asurement system	6
	3.1	Subsystems	6
	3.2	Thrust sensing unit	8
4	Exp	erimental validation	13
	4.1	Design of the plates	13
	4.2	Analysis of earlier measurements	15
	4.3	Analysis of distance sensor	16
	4.4	Analysis of the realised filter	17
5	Mea	asurements	19
	5.1	Axis definition	19
	5.2	Measurement setup	19
	5.3	Software	19
	5.4	Results	19
6	Con	iclusions	23
	6.1	Conclusions	23
	6.2	Discussions	23
	6.3	Recommendations and Future work	23
A	Mea	asurement software	25
B	For	ce/Torque sensor initialization of measurement software	29
С	Din	nensions of the plates	30

1 Introduction

1.1 Project background

SPECTORS is a project in which more than 31 companies and education facilities are cooperating. Their goal is to develop innovative hard- and software solutions to enhance competitiveness for small and medium-sized enterprises. The goal of the project is to exploit the civil drone technology market using improvements and innovations for civilian drones.[1] The Robotics and Mechatronics group of the University of Twente has a task within this project to develop an aerial robot which can achieve robust contact with a mechanical surface on a structure for inspection and interaction.

To have robust contact with a surface the aerial robot, or drone, needs to have a feedback of the exact force applied by the propeller on the drone's body. This information is also needed if the drone wants to apply a certain force on the surface.

Usually an open-loop controller is used for drone flight. the motion controller commands the rotor speed by assuming a predetermined map between the thrust produced and the rotor speed. This system has a disadvantage when a drone is flying with forces acting on its body different than assumed, for example flying close to a wall or in a windy environment.

For accurate position controls closed-loop controllers are preferred. Indoors high-speed optical sensing systems are used for accurate motion controls. Outdoors, however, these systems cannot be used. Also GPS is unreliable to have accurate motion control.[2]

A closed-loop controller for the thrust needs an accurate system that measures this force, otherwise the controller will not behave as intended.

1.2 Problem statement

To realise the robust contact with a surface, a closed-loop controller is preferred to get accurate measurements of the thrust acting on the drone. With this it can calculate its position and give the accurate force acting on the surface.

To realise the closed-loop controller an accurate thrust sensing unit needs to be devised.

1.3 Project goals

The first goal of my contribution will be to have a sensor system which accurately measures the force generated by the rotor.

To achieve this there are also secondary goals which will help to complete it:

The system must be able to filter the noise generated by the rotor for more accurate measurements of the thrust. Electronically filtering only helps to reduce the noise in the processing. However for this drone mechanical filtering is preferred as it reduces the vibrations of the drone, such that it can hover more steadily in a position. In the end if the mechanical filtering is not enough, electronical filtering can also be used.

The system will fit on the end of the beam together with the motor and rotor. In this way the system is not dependent on the drone and can act on its own. If this is possible it will be an independent thrust sensing unit with a motor and rotor included. The unit is able to be taken off the drone and connect with another drone if it meets the requirements.

Preferably the system is going to be small, so it will fit on the drone. This will help in a later stage, so it does not have to be made smaller. This also gives more challenges for every part hand-picked, because this has to be on a small scale and only commercial components are considered.

1.4 Report outline

This report consist of 6 chapters: The first chapter gives a general introduction of the assignment. Chapter 2 informs the reader about the background theory necessary to understand all the subjects treated. Chapter 3 explains the measurement setup and goes in on all the subsystems and choices made throughout the project. Chapter 4 goes in on the remaining parts to be designed and the analysis of an earlier measurement and different subsystems. Chapter 5 explains the measurement setup, software and also gives the results of the measurements. Chapter 6 gives the conclusion, discussion and recommended future work of this assignment.

2 Background

This chapter will discuss background aspects necessary for insight on the rest of the report. After reading this chapter everything should be clear for a reader with an electro technical background.

2.1 Aerodynamic forces

This section describes some aerodynamic forces acting on a wing or propeller. It will explain how lift is generated and briefly some unwanted forces.

2.1.1 lift

Wings or propeller's produce thrust or lift by 'pushing' the air in the direction perpendicular to the wings. The faster the wings move or rotate, the more air is pushed down and force upwards in generated. The amount of force generated is described by the lift equation[3]:

$$L = \frac{1}{2} * \rho * v^2 * A_{wing} * C_L$$

Where L is the lift force in Newton, ρ being the air density in kg/m^3 , v the velocity of the wing in m/s, A the wing area in m^2 and C_L the lift coefficient of the wing in arbitrary units. If a propeller is used, which rotates, the velocity of the wing will change in the rotating speed of the wing, ω , in m/s and the area of the wing changes in the area of the entire propeller over which pressure is formed by 'pushing' the air down. As can be seen the amount of lift is dependent on constants, and on the rotor velocity squared.

$$L = \frac{1}{2} * \rho * \omega^2 * A_{rot} * C_L$$

This equation assumes an equal amount of lift over the entire rotor. This, however, can only be assumed when the drone is not moving in a direction. When it does some unwanted forces will start to act.

2.1.2 Dis-symmetry of lift

The dis-symmetry of lift implies a larger amount of lift will be observed on the advancing blade than on the retreating blade. The advancing blade is the blade which is moving in the same direction as the drone is moving. The retreating blade is of course the other side. The advancing blade moves against the airspeed and thus has a higher relative airspeed than the retreating end. This produces a torque around the rotor centre, due to the different lift generated on both sides.

This phenomena is compensated in normal helicopters by another phenomena called blade flapping, which will be discussed later. On a drone with more than one propellers this is compensated by having an equal amount of propellers rotating in a clockwise direction as in the counter-clockwise direction.

2.1.3 Blade flapping

Due to the compensation of the dis-symmetry of lift on a drone, this phenomena is an unwanted side effect.

Due to the advancing blade having a higher relative airspeed, when the drone is moving, the blade will tilt slightly up and reduce the effective lift. On the other hand, the retreating blade will tilt less because it has a lower relative airspeed. This difference of lift will cause a torque around the axis. It is often referred to as the *rolling moment*.

2.2 BLDC motors and ESCs

This section is about the motor and control of the motor of a drone.

2.2.1 Brushless DC motors

Brushless DC (BLDC) motors are, because of their size and limited amount of space, one of the first choices on an UAV. BLDC motors are, in contrary to Brushed DC motors, motors without brushes, as seen in Figure 2.1. Brushes are mechanical inverters, also called a commutator, which generate an AC current to move the stator. BLDC motors use transistors to generate this AC current and so do not have an electrical connection between the moving rotor and the stationary hub. Because of this BLDC motors have a higher efficiency and higher mechanical wear at the cost of more complex controls.[4]

The rotational velocity is given by a 'KV rating'. This value gives the Rounds per Minute (RPM) of the motor at a certain input voltage without load. When a propeller is attached, the RPM will be reduced.



Figure 2.1: Brushed DC motor vs Brushless DC motor[5]

2.2.2 Electronic speed Controllers

To control a BLDC motor complex controls are necessary. This comes in the form of an Electronic Speed Controller (ESC). ESCs work by switching Field Effect Transistors (FET), on and off really fast. By adjusting the duty cycle of signal, the speed of the motor can be changed. ESCs generally create a three-phase AC signal with which the rotation of the motor can be arranged. By switching two of the wires between the ESC and BLDC motor, the rotation of the motor can be changed.[6]

An ESC is already chosen and is integrated in the measurement setup. The ESCs used currently in the project do not perform closed-loop control of the speed nor the thrust of the rotors.

2.3 Sensors and Wheatstone bridge

Below different type of sensors are listed which could be helpful for the setup. Also to read out some of the sensors a Wheatstone bridge is preferred. This will also be explained.

2.3.1 Resistive sensors

Strain gage

Strain gages are a type of sensor, but are also used in different kind of sensors, like pressure sensors. Due to the strain on an object, deformation, the 'wires' of the strain gage become longer, or shorter, and the resistance will change. This change can be measured with a Wheat-stone bridge.

linear sensors

Linear sensors are sensors which have a variable resistor or voltage which changes depending on the retraction of the rod. Usually they have a spring which pushes the rod to the utmost position and they have to be pushed in to get a change of value, voltage or resistor.

2.3.2 Optical sensors

Optical sensors are actually photo-diodes paired with a light-emitting diode (LED). The wavelength of the light produced by the LED circuit is usually near infra-red.[7] These components are placed with a certain angle next to each other, so the light emitted travels upwards and has to be reflected by some material, paper for example. This reflected light will be received by the photodiode and produce a current. The distance of the paper to the sensor is important, because the amount of reflected light determines the current. This current will be changed to a voltage, which can be read out.

2.3.3 Linear Variable Differential Transformers

Linear Variable Differential Transformers (LVDT) are linear sensors, but do not have a spring and they have coils, which determines the output voltage. This system uses an input voltage, Mostly AC, and runs it through a coil. Depending on the position of the rod, it will give an output voltage to one of the two coils placed as output. As the name suggests, this is a differential transformer.

2.3.4 Wheatstone bridge

A Wheatstone bridge is a setup which measures a voltage difference, depending on 1 or more variable resistors. The resistors are placed as seen in figure 2.2. All resistors have an equal value, but atleast one variable resistor will change in value. This causes a voltage difference between the two points, which can be read out.



Figure 2.2: Wheatstone bridge configuration[8]

3 Measurement system

An overview of the measurement system is given in Figure 3.1. This includes the thrust sensing system in parts. All parts will be discussed below. At last the analysis of the design choices of the thrust sensing system will be given. Due to the design of the mechanical filter and the choice of the sensor in parallel, there will be some cross references in the first section to the second.



Figure 3.1: Schematic of the measurement setup with thrust sensing system

3.1 Subsystems

3.1.1 PC

On the PC using the software pack MathWorks - MATLAB[9], The measurement data is collected in real-time. The sensor data is retrieved from the micro controller, while the data from the force/torque sensor is send to MATLAB directly. Further processing of the measurement data is also done on this PC, but not in real-time.

3.1.2 Microcontroller

The microcontroller is the most important part of the measurement setup. It controls the speed of the BLDC motor, by a GUI created in MATLAB, and transmits the sensor data back to the PC. A micro controller is already provided: This is the *Arduino Uno*[10]. The specifications are given in Table 3.1.

Specifications:	Arduino Uno
Operating Voltage:	5V
Input Voltage (recommended):	7-12V
Input Voltage (limit):	6-20V
Digital I/O Pins:	14 (of which 6 provide PWM output)
PWM Digital I/O Pins:	6
Analog Input Pins:	6
DC Current per I/O Pin:	20 mA
DC Current for 3.3V Pin:	50 mA
Length:	68.6 mm
Width:	53.4 mm
Weight:	25 g

Table 3.1:	Specific	ations	of Ardu	ino I	Jno
------------	----------	--------	---------	-------	-----

3.1.3 ESC

The electronic speed controller will be changing the rotation speed of the motor depending on the duty cycle of the PWM signal it receives. The electronic speed controller that is provided is the *DYS BL40A*[11]. The specification are in Table 3.2.

Specifications:	DYS BL40A	
Battery:	6-22.2 V (2-6S Lipo)	
Current Rating:	40amp 45a burst	
Signal frequency:	20-500 Hz	
Output PWM frequency:	18 KHz	
Size:	17x60x7.2 mm	
Weight:	20 g	

3.1.4 BLDC motor

The choice of a BLDC motor is important, because it depends together with the propeller the amount of thrust the drone can deliver. Also because it can bring about aerodynamic vibrations, which are not preferred. The BLDC motor that had already been chosen is the *Cobra CM2217/20*[12]. The specifications are in Table 3.3.

Specifications:	Cobra CM2217/20
Stator Diameter:	22.0 mm
Stator Thickness:	17.0 mm
Number of Stator Slots:	12
Number of Magnet Poles:	14
Motor Wind:	20 Turn Delta
Motor Kv Value:	950 RPM per Volt
No Load Current (Io):	0.53 Amps @ 12 Volts
Motor Resistance (Rm) per Phase:	0.188 Ohms
Motor Resistance (Rm) Phase to Phase:	0.125 Ohms
Maximum Continuous Current:	20 Amps
Max Continuous Power (3-cell Li-Po):	220 Watts
Max Continuous Power (4-cell Li-Po):	300 Watts
Motor Weight:	76 g
Outside Diameter:	27.7 mm
Motor Shaft Diameter:	3.17 mm
Prop Shaft Diameter:	5.00 mm
Motor Body Length:	33.0 mm
Overall Shaft Length:	35.1 mm
Motor Timing:	5-10 degrees
PWM Frequency:	8 KHz

Table 3.3: Specifications of Cobra CM2217/20

3.1.5 Propeller

A propeller has already been chosen. This is the *T-motor 11x3*.7[13]. The specifications for this propeller are in Table 3.4. As said before, the choice for a propeller is important, because it determines the amount of thrust it can deliver. It is also important for the efficiency of the BLDC. Too heavy of a propeller means the BLDC needs more power to generate the same amount of lift than a smaller one. The other way around ditto.

Specifications:	T-motor 11x3.7
Length:	11 inch
Pitch:	3.7 inch
Center Hole:	4mm
Outer Holes:	3mm
Outer Hole Centers:	12mm
Weight:	11.8 g
Rotation:	Clockwise and Counter-clockwise

Table 3.4: Specifications of T-motor 11x3.7

3.1.6 Force and Torque sensor

The Force/Torque sensor provided in the Robotics and Mechatronics lab is the *ATI Industrial Automation F/T Mini40E*[14]. The specifications of this sensor can be seen in Table 3.5. Care should be taken with mounting the thrust sensing system on the sensor, because too long screws can damage the sensor. Also care should be taken not to overstress the sensor. This can also damage the sensor permanently.

 Table 3.5:
 Specifications of ATI Industrial Automation F/T Mini40E

Specifications:	ATI Industrial Automation F/T Mini40E	
Maximum allowed axis force (x, y, z):	± 810, ± 810, ± 2400 N	
Maximum allowed axis torque (x, y, z):	± 19, ± 19, ± 20 Nm	
Maximum screw depth from surface:	5.0 mm	
Dimensions (diameter x height):	40 x 12.2 mm	
Weight:	49.9 gram	
Sample rate:	50Hz	

3.1.7 Power and Battery

To provide power to the micro controller and the ESC a battery was already provided. The micro controller will get power from a USB port on the PC and the ESC and motor will receive power from the *Delta Electronics SM 52-AR-60*[15] provided with the measurement cart. Its specifications are in Table 3.6.

Table 3.6:	Specifications	of Delta	Electronics	SM 52-AR-60
able 3.0.	specifications	UI Dena	Liectionics	5WI 52-AII-00

Specifications:	Delta Electronics SM 52-AR-60	
Maximum voltage per output	0-52 V	
Maximum current per output	0-60 A	
Maximum output power	1560 W	

3.2 Thrust sensing unit

3.2.1 Mechanical filter

The mechanical filter is going to be a mass-spring system, instead of a mass-spring-damper. This is due to the difficulty in obtaining off-the-shelf miniature dampers. The damping in the system will be only due to friction and inherent damping in the mechanical springs. The system will have a plate with the motor and propeller moving between springs. The top springs(1) will be used for the filter, because the force is generated upwards, and the bottom springs(2) will be used as shock damper. This choice will be discussed further. A sensor(3) below this plate will measure the height of the moving plate(4) to the sensor. Preferably the distance between the moving plate and the sensor is as small as possible. This system is seen in figure 3.2



Figure 3.2: Sketch of the preferred thrust sensing system

The mass of the moving plate is preferably as low as possible, because it has to fit on the drone at the end of a beam. The total mass of the moving plate is the mass of the material plus the motor and propeller. The masses of the motor and propeller provided are, 76 and 11,8 gram, respectively. The material used will be delrin, this can be arranged at the RaM faculty, it has a density of 1.41 g/cm^3 . The dimensions of the plates will follow, but the calculated weight of the material is 15 gram. This makes the total mass 102,8 gram.

The spring constant of the top springs depend on the maximum force delivered by the motor/propeller and on the distance they may be compressed. Due to force/torque sensor not being directly under the motor in the provided data, and still measuring a force of 14 Newton, it was decided to use 15 Newton as maximum force and that the moving plate should move maximum 5 mm. This results in a spring constant of 3 N/mm. Also because 4 springs will be working in parallel, this value will be divided by 4, so each spring will have a spring constant of 0.75 N/mm. These springs have been found and resulted in the D11010[16]. Due to the springs being 0.74 N/mm, the total spring constant is 2.96 N/mm.

The bottom springs are in the setup, due to a mistake at the time of ordering all components. At the time of ordering a different sensor was chosen, where it was necessary to have a minimal space between sensor and the moving plate. They will still be used as shock dampers. it will not affect the mechanical filter as the force is directed upwards. The springs on the bottom are the C0180-022-0310M[17].

Specifications:	D11010	C0180-022-0310M	
Outer diameter:	4.4 mm	4.57 mm	
Wire thickness:	0.4 m	0.56 mm	
Unloaded length:	11 mm	7.87 mm	
Loaded length:	4.1 mm	4.52 mm	
Maximum force:	5.25 N	13.92 N	
Spring constant:	0.74 N/mm	4.15 N/mm	

Table 3.7: Specifications of the top and bottom springs

3.2.2 Sensor choice

To measure the force generated by the rotors, the deflection of the plate will be measured by a distance sensor. The choice of a sensor was at first dependent on what distance it should

be able to measure with a good sensitivity. Because it is preferred the motor doesn't move too much, should be able to almost instantaneously lift the drone when mounted, it was determined that a distance of 5mm should be able to be measured.

To have the maximum thrust possible both plates have to be as small as possible. This also implies the sensor needs to be as small as possible.

A third requirement for the sensor is it should not be too expensive. Some sensors which had good specifications were too costly. This also made the list of possible sensors shorter.

The choice was made out of LVDTs, resistive sensors or optical sensors, because they had the best specifications for measuring a deflection.

LVDTs

Some LVDTs had good specifications. These were the *TE Connectivity MHR 250*[18], *TE Connectivity E 200*[19] and *TE Connectivity DC-SE 250*[20]. The specifications of these sensors are in table 3.8.

Specifications:	TE MHR 250	TE E 200	TE DC-SE 250
Input voltage:	3 VRMS sine wave	3 VRMS sine wave	8.5 to 28 VDC
Input current:	-	-	10mA maximum;
			6mA typical
Input frequency:	2 kHz to 20 kHz	50 Hz to 10 kHz	-
	(10 kHz recom-		
	mended)		
Stroke range:	± 6.35 mm	± 5.08 mm	6.35 mm
Sensitivity:	81.5 mV/V/mm	61.8 mV/V/mm	0.787 VDC/mm
Output at stroke ends:	517.5 mV/V	314 mV/V	-
Non-linearity @ 100%	0.25%	$\pm 0.5\%$	0.25%
stroke (max):			
Body length:	47 mm	57.2 mm	110.7 mm
Body weight:	9 gram	36 gram	91 gram

Table 3.8:	Specifications	of the LVDTs
14010 0101	opeemicationo	or the Lt D to

Also the *TE Connectivity MHR 100* and the *TE Connectivity E 100*, the smaller versions of the preferred sensors, fall within the requirement of a measurement distance of 5mm, but then there would be no room for error and these sensors still have a high price. These sensors were taken out of consideration.

Resistive sensors

Two resistive sensors were also looked after: The *BEI* 9605[21] and the *Vishay 20LHE*[22]. The specification of these sensors are in table 3.9.

Specifications:	BEI 9605	Vishay 20LHE
Input voltage:	-	5 VDC ± 10%
Input current:	-	< 16 mA typical
Output signal:	-	Analog ratiometric 10%
		to 90% of V_{input} or PWM
		10% to 90% duty cycle
Electrical stroke:	12.7 mm	up to 10 mm
Mechanical stroke:	14.2 mm	12 mm
Linearity:	2%	± 2%
Total DC Resistance \pm 25%:	1.7 ΚΩ	-
Housing length:	26.9 mm	-
Rod length (fully extended):	20.6 mm	-
Dimensions:	-	46 x 20.8 x 37 mm
Weight:	-	26 gram
Actuation force:	4 N	1.5 to 7 N typical

Table 3.9: Specifications of the BEI 9605 and Vishay 20LHE

Optical sensors

Optical sensors were looked at. Generally they are the best option for distance measurements without adding forces to the system.

Only for a distance of 5mm a few sensors are left: the *Sharp GP2Y0A41SK0F*[23] and the *Sharp GP2Y0A51SK0F*[24]. The specifications of these sensors are in table 3.10

Specifications:	Sharp GP2Y0A41SK0F	Sharp GP2Y0A51SK0F
measuring distance:	4 to 30 cm	2 to 15 cm
Input voltage:	4.5 to 5.5 V	4.5 to 5.5 V
Input current:	22 mA maximum; 12 mA typical	22 mA maximum; 12 mA typical
Output terminal voltage:	-0.3 to <i>V</i> _{input} +0.3	-0.3 to <i>V</i> _{input} +0.3

Table 3.10: Specifications of Sharp GP2Y0A41SK0F and GP2Y0A51SK0F

Discussion

The LVDTs are plausible sensors for measurements on small scale, but the problem with these sensors is their size and price. The size problem is their body length. They should be mounted on top of the moving plate, or on the bottom plate. If placed on the top plate, the body length of the sensor should not be more than the height of the motor, which is 33mm. On the bottom is also hard to mount, because this system could be mounted on the drone on the bottom. Also mounting it in the middle, between the two plates, is undesirable: the distance between the two plates would become a lot more.

The mechanical sensors have one disadvantage: they have an internal spring to keep the beam in the utmost position. This spring adds another force to the filter, which is not preferred. Also the length of the Vishay 20LHE is too long to put on the top of the moving plate. The BEI 9605 has a smaller length, but adds more force to the filter. For the placement on the system, because it adds a force, preferably it is placed in the middle. Only the motor is already in the middle, so this is not an option. Placing in the middle of the bottom plate is possible, but then the system will be much higher. Another option is to place more sensors around the motor, but then it will add more force to the system, which could become a problem.

Optical sensor have the advantage they do not interfere with the setup. The disadvantage on the other hand is they need to be placed on a distance of the setup. The 2 to 15 cm seems as the better sensor, because they have the same specifications except the measuring distance.

The most important aspect of the sensor is it should measure accurately with a good sensitivity, because it is a small distance, and secondly it is beneficial if it does not add extra forces. Due to these specifications the LVDTs are the best choice, but they are too costly for this project. The distance is also a disadvantage, but this is also a problem with the second best sensor: the optical sensor. It can be used if a solution is made for the distance. This will be done by adding some height to the sensor.

4 Experimental validation

With all choices made on the measurement setup, the thrust sensing unit can be seen in figure 4.1. Only the design of the plates remains. This will be done below, because they were designed to meet the requirements after all choices were made.



Figure 4.1: Measurement setup

4.1 Design of the plates

At last the two plates where the sensor and the motor will be attached to, have to be designed. The moving plate, as seen in figure 4.2, has 4 holes in the middle in a 16x19 mm pattern for the M3 screws included with the motor. It also needs 4 holes for the M4 screws which will be attached to the bottom plate. Around these screws the springs will be placed; one above and one below.



Figure 4.2: Top plate for the setup

The bottom plate has the 4 M4 holes for the screws. Only these holes are 3.1mm instead of 4, because screw thread will be fabricated to fasten the screws. Also it needs 3 holes for the force/torque sensor, also with screw thread to anchor the bottom plate on the sensor. This reduces the vibration by not being tightened enough. This plate also has 2 holes for the optical sensor, which are placed off centre, because otherwise the screws will not fit next to each other. This plate can be seen in figure 4.3.



Figure 4.3: Bottom plate for the setup

Both plates are preferably as small as possible, because the air pushed down by the propeller will flow against the plates and try to push it down. This causes the total thrust of the motor and propeller to decrease. To not be able to bend the plate, the thickness will be 3 mm. Also a

small white paper piece is put underneath the top plate for a better reflection of the light of the sensor.

Due to the sensor distance being 4 cm, the screws that hold the top plate are placed on spacers of 50mm long. Also M3 rings, with M4 screw thread, were placed between the top springs and the plate. Otherwise the springs would drop inside the hole of the plate.

The dimensions for the total setup, without propeller are: width and length 60 mm and height 129mm. The measured weight of the top plate with motor and propeller is 92 gram. The exact dimensions of the plates are in Appendix C

4.2 Analysis of earlier measurements

A measurement was performed before the start of the assignment which had the data of the total thrust and torque generated by the motor and propeller. This data can be seen in figure 4.4 and table 4.1.

The maximum thrust of this measurement was used for the design of the mechanical filter and this measurement will be used as comparison for the measurement results.

In the table 11 percent is used, instead of 10, because the motor was not rotating until 11 percent. The average force was calculated for every percentage by taking the mean of all the sample points where it was any percentage of the PWM. Of every level the first second was distracted, because the motor could still have been accelerating in that time.

The maximum thrust of this measurement is around 12.5 N, but for the design of the mechanical filter a value of 15 N was used to have some tolerance.

A second degree polynomial fit is made from the data of the table, such that it can be compared to the force vs percentage of the PWM signal data obtained from the measurements. The equation of the second degree polynomial is: $y = 7.7760e^{-4} * x^2 + 0.0636 * x - 0.5586$ with y being the generated force and x the percentage of the PWM signal.



Figure 4.4: Measurement data of the test performed earlier

PWM signal (%)	Force (N)
0	0
11	0.2
20	0.8
30	1.6
40	2.8
50	4.4
60	6.2
70	8.2
80	10.1
90	12.1
100	12.5

Table 4.1: Amount of force generated vs percentage of PWM signal of the earlier measurement

4.3 Analysis of distance sensor

The output voltage relating to the distance can be seen in figure 4.5. This data was estimated, because only the graph was provided. For the estimation a first degree polynomial fit was made for the data between 4 and 6 cm. The distance measured will be between 4.5 and 5 cm, due to the mechanical construction. The points 4 to 6 cm are taken to get a more accurate prediction. This will be used to convert the measurement result of the voltage output to the distance the plate moved. The equation of the first degree polynomial fit is: y = -0.4314 * x + 4.4843 with y being the voltage and x the distance between the top plate and the sensor.



Figure 4.5: Output voltage vs Distance for the optical sensor[23]

4.4 Analysis of the realised filter

The mechanical filter is also simulated in MATLAB. This is done with the following assumptions:

- Top plate only moves in Z-axis
- Top plate does not bend
- No deviation between spring constants
- springs add no significant mass
- Motor shaft only rotates, no secondary movements
- Connection between top plate and screws does not produce friction

With these assumptions the mechanical filter is a simple mass-spring system with the transfer function: $H(s) = \frac{1}{s^2 * m + K}$, wherein m is the mass of 0.092 kg and K is the spring constant of 2960 N/m. The bode plot of this filter is seen in figure 4.6.

When this system is attached to a drone, the bottom plate will not be solid to the ground, but also fly with a mass attached to it. The system will behave differently due to the mass. This was not analysed and simulated due to time constraints and will be a future recommendation.



Figure 4.6: Bode plot of the mechanical filter

This distance can be related to an output voltage with the help of the polynomial function made from the manufacturer data[23]. The expected voltage output will be between 2.76 and 2.56V for a distance of 4.5cm and up. Also a polynomial fit is made to compare it with the measurement data. This equation is: y = -0.0144 * x + 2.5430 with y being the output voltage and x the generated force

Due to secondary aerodynamic effects[25], this force will not be singly in the Z-axis, such that the top plate not only moves in the Z-axis, but will also generate some force in the X and Y directions.

5 Measurements

5.1 Axis definition

The axis are defined as follow:

The positive X-axis is in the direction of the beam of the measurement setup, also where all the cables are going, the positive Y-axis is to the left of the X-axis, 90 degrees counter clockwise, and the positive Z-axis is upwards. The measurement setup for the axis definition can be seen in figure 4.1.

The torque for every axis is the counter-clockwise rotation in the positive direction.

5.2 Measurement setup

The measurements were done with a similar setup described in the previous mentioned report[25]. The only differences were the power supply used to power the motor and the distribution board, which now had an Arduino Uno instead of the MCU. Also the thrust sensing system was placed on the arm of the cart without an extra arm, as seen in figure 4.1

5.3 Software

The software used to execute the measurements is provided. It uses a GUI to control the motor speed, between 0 and 100 percent, and to start and stop the measurement. The force/torque sensor is connected to MATLAB using a LAN connection and the output values can be seen in Simulink. This file had to be modified to read out the voltage of the optical sensor, which is connected and powered by the Arduino Uno. Two buttons were made on the GUI to start and stop the sensor readout. This could not be read out continuously, because when it was interrupted by another action, for example increasing the motor speed, the sensor readout stopped. This file and a figure of the GUI is included in Appendix A and B.

The measurement is executed by increasing the motor speed to a certain percentage, for this measurement 10 to 100 percent with increments of 10, then start the sensor readout for 5 to 10 seconds. When this is done, increase the motor speed to the next level and repeat.

5.4 Results

The first measurement shown is the force produced, the voltage of the sensor and the input PWM signal. The first thing to see is that the motor and/or filter produces around 0.5 N of variations when it has a steady input signal. Also the maximum input signal is 50 percent. This is also due to enormous vibrations starting at 54 and later at 60 percent. These noises are discussed later in the report.

The connection of the force/torque measurement happens via LAN, it can be seen that the signal lags around 2.5 seconds. The rest of the figures is made by processing this time delay.

The mean force produced due to PWM signal is seen in the next table. This is done by taking the time where the input signal is steady on a percentage, in this measurement: 11, 20, 30, 40 and 50 percent. The first second of every measurement is subtracted, because the motor could still have been accelerating.

A polynomial fit was made for this data in comparison with the provided data in figure 5.2. until 50 percent the amount of force generated closely relates to the provided data.



Figure 5.1: Measurement data of the input PWM percentage, Force produced and sensor Voltage

PWM Signal (%)	Force (N)
0	0
11	0.2
20	0.8
30	1.7
40	3.0
50	4.6





Figure 5.2: Polynomial functions of data provided and measurement data

Figure 5.3 shows the output voltage of the sensor vs the force. As can be seen the voltage has a large variation for the different levels of force. This is due to the vibrations of the plate mentioned earlier. This will be discussed later. One measurement goes above the axis. This measurement was 3.1 V and can also be seen in figure 5.1. It is probably a faulty measurement. Due to the large swing of the voltage it cannot be seen if the sensor actually measures the plate going up. Taking the average voltage for a force can show if the sensor detects it. This has to be taken with caution however, because there is too much noise. With the polynomial fit it can be seen that the voltage output is indeed going down for the mean values of every PWM percentage and that it closely relates to the manufacturer data.



Figure 5.3: Output voltage for different levels of Force

Using the manufacturer data and spring constant to relate the voltage of the sensor in the measurements to an amount of force can be seen in figure 5.4. The actual force can be seen against the expected force due to the sensor voltage. It shows this setup cannot accurately measure the force. This is probably due to the amount of noise generated because of the vibrations.



Figure 5.4: Calculated force output vs measured force output

The forces in the X and Y axis from the measurements are compared to the forces in the provided data in figure 5.5. They are plotted against the PWM signal. The forces in the X- and Y-axis are in some regions bigger and in some smaller than the provided data. This can be

interpreted as this system behaving the same as the setup from the provided data. In the Z-axis the difference is harder too see, because it produces a different force than the provided data. However in some regions where the lines are split, it looks like it produces the same noise as the provided data.



Figure 5.5: Forces X- and Y-axis of measurement and earlier test data

The heavy vibrations mentioned earlier can be seen in the video included with this report. Some snapshots were taken from the video to show what happened in figure 5.6:



Figure 5.6: Heavy vibrations of the measurement setup

The top plate started to vibrate around an input PWM signal of around 54 percent. Those vibrations were too massive to continue the measurements above 50 percent. The choice was made to only use measurements until 50 percent, due to this problem.

As can be seen in the video it looks like the spacers start to resonate with the vibrations and makes the whole setup move more violently. This has to be analysed to find the cause and improve the system in such a way that it is possible to measure above 50 percent.

6 Conclusions

6.1 Conclusions

A thrust sensing system was designed which is able sense an amount of thrust, but not yet accurately. This is due to the vibrations generated by the motor and propeller.

It is a miniature design which fits on the end of the beam and it can act on its own as a small gadget. Only the height is not preferred, but this is due to the sensor which cannot measure the distance accurate with smaller space.

The filter which was preferred could not really act as a filter, because it is a mass-spring system. These systems are designed to have a sensing range, until their resonance frequency, and after they reached the peak of the resonance, they filter out every other frequency. To get more filtering it is preferred to have a damper with the mass-spring system.

After taking the average of the voltage output the outcome shows potential to become a good thrust sensing system. It still has some disadvantages, such as:

- Able to move in more than one direction
- Too much voltage swing in the output
- Setup too high, which can cause more backlash

The output of the sensor was later also measured when the top plate is in a steady position. The voltage still had some differences, but the vibrations caused by the motor and propeller caused the most fluctuation in the output.

6.2 Discussions

one way to improve this system is to find a sensor which can measure the distance on a smaller scale. This would drastically decrease the total height of the system. It could also help to reduce the heavy vibrations produced at higher input PWM %, because the height could be a problem. This has to be investigated though.

The transfer function of the system could not be measured, because it was not reliable to let the system go from 0 to 100 percnet in a really short time. This could be done in the future when the setup is reliable enough.

6.3 Recommendations and Future work

With this setup due to not being able to go higher than 50 percent, a recommendation is to analyse what goes wrong and improve it, so more tests can be concluded.

To have less noise in the X- and Y-axis the system can be improved to not be able to rotate around, so it will only generate thrust in the Z-axis.

A better sensor can be tried to find, because to have a precise representation of the force through the distance measurement, it is still not good enough.

This thrust sensing system could be compared to a current measurement to see which system performs better.

If a way could be found to reduce the vibrations during the test, which caused the high voltage fluctuations, the force could be measured more accurately even with the sensor already chosen.

As said in the discussion, when the setup is reliable enough to go from 0 to 100 in an instant, it is possible to accurately identify the actual transfer function of the mechanical system. Also the MATLAB file included for the measurements is not suitable to do a step function test. This

also has to be improved for the transfer function to be measured. It is necessary to know the accurate transfer function, because this will be used in the closed-loop controller. Otherwise the total system will still calculate the amount of thrust wrongfully.

Due to the Force/Torque sensor having a sampling rate of 200 Hz, the aerodynamic noise generated by the motor and propeller can hardly be analysed. The frequency generated is generally above 100 Hz. If a sensor can be used which has a higher sampling frequency, this noise can be analysed.

If the aerodynamic noise can be analysed, the filter can be characterised better, so it will be more resistant to all the vibrations.

During the test, the connection between MATLAB and the force/torque sensor had a small delay before MATLAB received the data. To make the processing of the measurement results easier and more precise, it could be investigated where this delay comes from: network issues or maybe hardware dependent.

A Measurement software

```
1 function varargout = GUI_v3(varargin)
<sup>2</sup> % GUI_V3 MATLAB code for GUI_v3.fig
3 %
4 % Begin initialization code - DO NOT EDIT
5 gui_Singleton = 1;
  gui_State = struct('gui_Name',
6
                                          mfilename, ...
                        'gui_Singleton', gui_Singleton, ...
7
                        'gui_OpeningFcn', @GUI_v3_OpeningFcn, ...
8
                        gui_OutputFcn', @GUI_v3_OutputFcn, ...
9
                        'gui_LayoutFcn', [] , ...
10
                        'gui Callback',
                                          []);
11
   if nargin && ischar(varargin {1})
12
       gui_State.gui_Callback = str2func(varargin {1});
13
  end
14
15
   if nargout
16
       [varargout {1:nargout}] = gui_mainfcn(gui_State, varargin {:});
17
18
   else
       gui_mainfcn(gui_State, varargin{:});
19
  end
20
  % End initialization code – DO NOT EDIT
21
22
23 % ---- Executes just before GUI_v3 is made visible.
<sup>24</sup> function GUI v3 OpeningFcn(hObject, eventdata, handles, varargin)
25 % This function has no output args, see OutputFcn.
  % hObject
               handle to figure
26
27 % eventdata reserved – to be defined in a future version of MATLAB
                structure with handles and user data (see GUIDATA)
  % handles
28
<sup>29</sup> % varargin command line arguments to GUI_v3 (see VARARGIN)
  clc
30
  % Choose default command line output for GUI_v3
31
  handles.output = hObject;
32
33
<sup>34</sup> % Update handles structure
  guidata(hObject, handles);
35
36
  global ard serv sensortimer
37
38
  ard = arduino();%'COM3', 'UNO', 'Libraries', 'Servo')%('COM6', 'Uno
39
       ', 'Libraries', 'Servo')
40
41 % DYS ESC BL-HELI Flashed
  serv = servo(ard, 'D6', 'MinPulseDuration', 1000*10^-6, '
42
      MaxPulseDuration', 2000*10^-6)
43
44 % Maytech ESC
  %serv = servo(ard, 'D6', 'MinPulseDuration', 1067*10^-6, '
45
      MaxPulseDuration', 1860*10^{-6})
```

```
<sup>46</sup> writePosition(serv, 0);
  init_FT_sensor;
47
  open_system('Simulink_Interface_Test_1')
48
49
_{50} % --- Outputs from this function are returned to the command line.
  function varargout = GUI_v3_OutputFcn(hObject, eventdata, handles)
51
 % varargout cell array for returning output args (see VARARGOUT);
52
 % hObject handle to figure
53
54 % eventdata reserved – to be defined in a future version of MATLAB
55 % handles
                structure with handles and user data (see GUIDATA)
56
  % Get default command line output from handles structure
57
  varargout{1} = handles.output;
58
59
  function testConn_Callback(hObject, eventdata, handles)
60
  global ard
61
  for i = 1:5
62
         writeDigitalPin(ard, 'D13', 0);
63
         pause(0.25);
64
         writeDigitalPin(ard, 'D13', 1);
65
         pause(0.25);
66
  end
67
68
  function pwmValueSlide_Callback(hObject, eventdata, handles)
69
  global serv
70
  pwmVal = get(hObject, 'Value')
71
  set (handles.pwmValueDisp, 'String',num2str(pwmMap(pwmVal)));
72
  writePWM (pwmVal)
73
74
  function pwmValueSlide_CreateFcn(hObject, eventdata, handles)
75
  if isequal(get(hObject, 'BackgroundColor'), get(0, '
76
      defaultUicontrolBackgroundColor'))
       set(hObject, 'BackgroundColor',[.9 .9 .9]);
77
  end
78
79
  function pwmValueDisp_Callback(hObject, eventdata, handles)
80
  global serv
81
82 pwmValInv = str2double(get(hObject, 'String'))
  pwmVal=invPwmMap(pwmValInv)
83
  set (handles.pwmValueSlide, 'Value',pwmVal);
84
  writePWM (pwmVal)
85
86
  function pwmValueDisp_CreateFcn(hObject, eventdata, handles)
87
   if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, '
88
      defaultUicontrolBackgroundColor'))
       set(hObject, 'BackgroundColor', 'white');
89
  end
90
91
<sup>92</sup> % ---- Executes when user attempts to close figure1.
  function figure1_CloseRequestFcn(hObject, eventdata, handles)
93
  delete (hObject)
94
```

```
95 clear all
96
97 function v^2 = pwmMap(v^1)
_{98} %a = 35;
99 \% b = 65;
100 a = 0;
b = 100;
102 \quad c = 0;
d = 100;
104
  v_2 = ((c+d)+(d-c)*((2*v_1-(a+b))/(b-a)))/2;
105
106
107 function v2 = invPwmMap(v1)
108 \ \% C = 35;
109 \ \% d = 65;
110 C = 0;
d = 100;
112
113 a = 0;
b = 100;
115
116 v2 = ((c+d)+(d-c)*((2*v1-(a+b))/(b-a)))/2;
117
<sup>118</sup> function writePWM(pwmVal)
119 global serv
writePosition(serv, pwmVal/100.0);
121 current_pos = readPosition(serv)
   set_param('Simulink_Interface_Test_1/pwmValue', 'Value', num2str(
122
       pwmMap(pwmVal)));
123
<sup>124</sup> % ---- Executes on button press in stopSimulink.
<sup>125</sup> function stopSimulink_Callback(hObject, eventdata, handles)
set_param('Simulink_Interface_Test_1', 'SimulationCommand', 'stop')
127 global sensortimer
  delete (sensortimer)
128
  writePWM (invPwmMap(0))
129
130
131 % --- Executes on button press in startSimulation.
<sup>132</sup> function startSimulation_Callback(hObject, eventdata, handles)
  set_param('Simulink_Interface_Test_1', 'SimulationCommand', 'start')
133
   writePWM(invPwmMap(0))
134
135
136 % --- Executes on button press in stopsensor.
<sup>137</sup> function stopsensor_Callback(hObject, eventdata, handles)
138 global sensortimer
<sup>139</sup> delete (sensortimer);
140 set_param('Simulink_Interface_Test_1/sensorValue','Value',num2str
       (0));
141
142 % ---- Executes on button press in startsensor.
  function startsensor_Callback(hObject, eventdata, handles)
143
```



Figure A.1: User interface of the measurement software

B Force/Torque sensor initialization of measurement software

```
udp_port = udp('192.168.1.1', 49152, 'Localport', 40000); % open
1
      udp_port
<sup>2</sup> header = 4660; % required
3
_4 mode = 2; % 2 = continue sending packets
  sample_count = 0; % 0 = infinity
5
6
7 % swapbytes and define type
8 header = swapbytes(uint16(header));
9 mode = swapbytes(uint16(mode));
  sample_count = swapbytes(uint32(sample_count));
10
11
12 % convert it to array of uint8 (chars)
13 header = typecast(header, 'uint8');
14 mode = typecast(mode, 'uint8');
  sample_count = typecast(sample_count, 'uint8');
15
16
17 packet = [header mode sample_count];
18 fopen(udp_port);
19 fwrite (udp_port, packet);
  fclose(udp_port);
20
21
22 clear udp_port packet;
```



C Dimensions of the plates

Bibliography

- [1] Spectors, "Sensing the World." [Online]. Available: https://spectors.eu/wordpress
- [2] E. Davis, P. E. I. Pounds, *Direct Sensing of Thrust and Velocity for a Quadrotor Rotor Array*, Jun. 2017.
- [3] MIT Department of Aeronautics and Astronautics, "Theory of flight." [Online]. Available: http://web.mit.edu/16.00/www/aec/flight.html
- [4] Wikimedia, "Brushless dc electric motor." [Online]. Available: https://en.wikipedia.org/ wiki/Brushless_DC_electric_motor
- [5] Think RC, "Brushed vs brushless motors." [Online]. Available: http://www.thinkrc.com/ faq/brushless-motors.php
- [6] T. Agarwal, "Introduction То Electronic Speed Control (ESC) Working and Applications." [Online]. Available: https://www.elprocus.com/ electronic-speed-control-esc-working-applications/
- [7] J. S. Wilson., Sensor Technology Handbook, J. S. Wilson, Ed. Elsevier, 2015.
- [8] Tangient LLC, "Wheatstone bridge configuration." [Online]. Available: https://takvwo. wikispaces.com/Wheatstone+brugschakeling.
- [9] Mathworks, "Matlab." [Online]. Available: https://nl.mathworks.com/products/matlab. html
- [10] Arduino, "Arduino Uno." [Online]. Available: https://store.arduino.cc/arduino-uno-rev3
- [11] DYS, "BL40A BLHeli." [Online]. Available: https://www.flyingtech.co.uk/electronics/ dys-bl40a-blheli-multirotor-2-6s-opto-40a-mini-esc
- [12] Cobra, "CM2217/20." [Online]. Available: http://innov8tivedesigns.com/ cobra-cm-2217-20-brushless-motor-kv-950
- [13] T-Motor, "11x3.7 prop." [Online]. Available: https://hobbyking.com/en_us/ multirotor-carbon-fiber-t-style-propeller-11x3-7-black-cw-ccw-2pcs.html
- [14] ATI Industrial Automation, "F/T sensor: Mini40E." [Online]. Available: http://www.ati-ia. com/products/ft/ft_models.aspx?id=Mini40
- [15] Delta Electronics, "SM 52-AR-60." [Online]. Available: http://www.delta-elektronika.nl/ en/products/dc-power-supplies-1500w-sm1500-series.html)
- [16] Amatec, "D11010 spring." [Online]. Available: https://www.amatec.nl/webshop/ drukveren/D11010
- [17] —, "C0180-022-0310M spring." [Online]. Available: https://www.amatec.nl/webshop/ drukveren/C0180-022-0310M
- [18] TE Connectivity, "MHR-series." [Online]. Available: http://www.te.com/commerce/ DocumentDelivery/DDEController?Action=srchrtrv&DocNm=MHR&DocType=Data+ Sheet&DocLang=English
- [19] —, "E-series." [Online]. Available: https://docs-emea.rs-online.com/webdocs/1308/ 0900766b81308356.pdf

- [20] —, "DC-SE-series." [Online]. Available: http://www.te.com/ commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=DC_SE_ General-purpose-DC-LVDT&DocType=DS&DocLang=English
- [21] BEI Sensors, "9600 series." [Online]. Available: http://ecatalog.beisensors.com/Asset/ linear-position-sensor_9600.pdf
- [22] Vishay, "20LHE series." [Online]. Available: http://www.vishay.com/docs/57115/20lhe. pdf
- [23] Sharp, "GP2Y0A41SK0F." [Online]. Available: https://www.pololu.com/file/0J713/ GP2Y0A41SK0F.pdf
- [24] —, "GP2Y0A51SK0F." [Online]. Available: https://www.pololu.com/file/0J845/ GP2Y0A51SK0F.pdf
- [25] B. Okken, Design and Empirical study of tilting propellers of an over-actuated quadrotor, Jul. 2017. [Online]. Available: https://www.ram.ewi.utwente.nl/aigaion/attachments/ single/1387