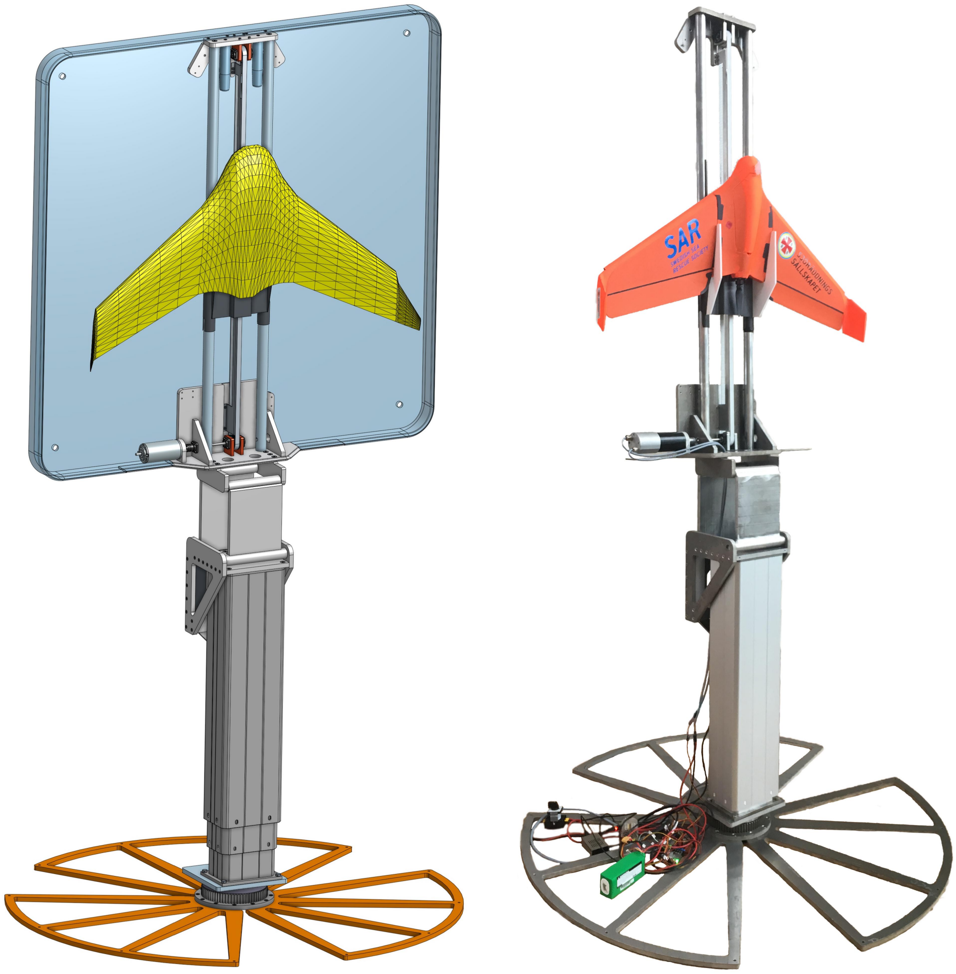
CHALMERS
UNIVERSITY OF TECHNOLOGY



Design and mechatronic integration of a drone launcher

Master's thesis in Systems, Control and Mechatronics

DANIEL VALERO BELTRÁ

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018

Design and mechatronic integration of a drone launcher

DANIEL VALERO BELTRÁ

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2018

Design and mechatronic integration of a drone launcher
DANIEL VALERO BELTRÁ

Supervisor: Boaz Ash, Department of Electrical Engineering, and Fredrik Falkman, Swedish Sea Rescue Society
Examiner:Petter Falkman, Department of Electrical Engineering

Master's Thesis 2018
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Isometric view of the drone launcher CAD's model designed with *Onshape* and the real prototype.

Typeset in LaTeX
Gothenburg, Sweden 2018

Design and mechatronic integration of a drone launcher
DANIEL VALERO BELTRÁ
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis is part of a drone launcher project developed by the Swedish Sea Rescue Society (SSRS). The objective of the project is to have a fixed drone launcher that could be installed next to the coast to be ready to send a drone anytime it is needed. The drone will be used to monitor and to gather information during sea rescue missions.

The upper capsule where the drone is installed and its launching mechanism have already been designed, but this capsule needs to move to be able to make a successful launch. There are three necessary movements that need to be implemented: a vertical displacement to increase and reduce height, a rotation to establish the direction of the launch and a pitch that modifies the angle with the ground, depending on the wind.

To fulfil these requirements, different types of mechanisms were designed in this thesis. They were analysed to test their viability and, once one is chosen, a working prototype was built based on it. The control system implemented is based on a Raspberry Pi and two motor controllers, used to control a linear actuator, a lifting column and two DC motors. In order to control the position, encoders were used.

In the next iterations of the project, a communication platform between various drone launchers needs to be implemented, also the mechanism for the capsule aperture has to be designed, and it is possible to integrate solar cells for powering the launcher.

Keywords: mechatronics, drone, launcher, linear actuator, control, Flask, Swedish Sea Rescue Society.

Acknowledgements

First, I would like to thank Petter Falkman for offering me a very interesting project, even though I was an exchange student. To Boaz Ash for his invaluable advices and for always recieving me with a smile every time I went to the office. I would also like to acknowledge Fredrik Falkman for his help, his trust and for showing me what a real engineer can do just with a blank paper and a pen.
Besides, I would like to remember my friends from the exchange, Lotte, Jaakko, Patrick, François, Roberta and Victor, whose friendship made my semester much more interesting. Finally, I would like to thank my parents for giving me the opportunity to study abroad and do something as crazy as this.

Daniel Valero Beltrá, Gothenburg, September 2018

Contents

List of Figures

List of Tables

List of Symbols

α_{max}	Maximum angular acceleration
β	Column effective length factor
μ	Friction coefficient
$\omega_{\text{launch final}}$	Final angular speed of the launch motor
ω_{launch}	Necessary angular speed of the launch motor
ρ_{wind}	Wind density
a	Acceleration
A_{capsule}	Area of the capsule
A_{PE}	Polyethylene area
C	Dynamic load of the bearing
C_0	Static load of the bearing
C_{battery}	Discharge rate of the battery
C_{d}	Drag coefficient
D_{actuator}	Distance between actuator and column
d_{bracket}	Distance between actuator and bracket
d_{launch}	Launch distance
d_{mass}	Mass centre distance
D_{pw}	Average between the inside diameter and the outside diameter of the bearing
d_{wind}	Distance between column and wind load
E	Aluminium Young's modulus
F_{actuator}	Actuator force
F_{a}	Axial load on the bearing
F_{friction}	Friction force
$F_{\text{horizontal}}$	Horizontal load
F_{launch}	Launch force
F_{r}	Radial load on the bearing
F_{s}	Safety factor of the bearing
F_{wind}	Wind force
g	Standard gravity
h	Height of the capsule
$I_{\text{belt gears}}$	Belt gears gear ratio
I_{capsule}	Moment of inertia of the capsule
I_{launch}	Launch current
$i_{\text{p gearbox}}$	Pitch gearbox gear ratio
$i_{\text{r gearbox}}$	Rotation gearbox gear ratio
I_{slider}	Slider area moment of inertia

i_{worm}	Worm gearbox gear ratio
L	Beam length
L_{bearing}	Basic life rating
M_{actuator}	Actuator momentum
M_{bearing}	Momentum on the bearing
M_{bracket}	Bracket momentum
m_{capsule}	Capsule mass
$M_{\text{drone sled}}$	Drone and sled mass
M_{f}	Final rotation torque
$M_{\text{max bearing}}$	Maximum momentum supported by the bearing
$M_{\text{max gearbox}}$	Maximum input torque Maxon gearbox
M_{motor}	Maxon motor nominal torque
M_{required}	Worm gearbox required momentum
M_{wind}	Momentum due the wind
$M_{\text{worm f}}$	Worm gearbox final torque
$M_{\text{worm i}}$	Worm gearbox input torque
$M_{\text{worm o}}$	Worm gearbox output torque
N	Normal force
n_{motor}	Maxon motor nominal speed
P	Dynamic equivalent load of the bearing
P_0	Static equivalent load of the bearing
P_{cr}	Critical load
P_{launch}	Launch power
P_{PE}	Polyethylene pressure
R	Reaction
$r_{\text{belt launch}}$	Radius of the wheel driving the belt
T_{launch}	Launch time
$T_{\text{pitch 90}}$	90°pitch time
T_{rotation}	Rotation time
V_0	Initial speed
V_{f}	Launch speed
V_{launch}	Launch motor voltage
V_{pitch}	Pitch speed
V_{rotation}	Rotation speed
V_{wind}	Wind speed
w	Width of the capsule
W_{launch}	Launch work
w_{slider}	Slider width
X	Radial load coefficient of the bearing
Y	Axial load coefficient of the bearing
y	Displacement

1
Introduction

1.1 Background

In the last years, an important growth on the number of developments focused on drones has been experienced. With the maturity of these technologies new usages for the drones have appeared and nowadays it is possible to see them in many different tasks.

The Swedish Sea Rescue Society (SSRS) is an organization focused on search and rescue missions all over Sweden. Since it does not receive any money from the government, all the operations are financed by its members and they are possible thanks to the work of the volunteers. Despite this fact, the society is responsible for more than 70 percent of all sea rescues in Sweden, covering most of the rescues except for the ones where another equipment, like a helicopter, is needed.

The intention of SSRS with this project is to increase their possibilities during sea rescues by launching a drone before the arrival of the rescue team. The drone, equipped with several sensors and a camera, could record a video that could be broadcast to all the stations. With this video, the team could be capable of evaluating in a more appropriate way the cause and urgency of the situation, and also be able to monitor the whole operation from the air.

But the drone and its launcher can have other purposes than just being used in the rescue missions. With a good design and presentation, they could also be used as an advertisement for the organisation, helping to gather more donations, increasing the number of volunteers and even showing the society the continuous innovations done by their team to deliver a better service.

However, the idea of using drones for sea rescue is not new and it is just a result of the proliferation of drones to different areas. In the scenario of sea rescue missions, it makes sense to use drones because of their agility, autonomous behaviour and capability to mount different sensors. Another critical point is the general acceptance that they have in the society. A survey showed that 88 percent of civilian accept the usage of drones in search and rescue missions [1].

In some countries, like Belgium, the Royal Military Academy developed a project named ICARUS that gives robotic support for marine search and rescue operations. In this programme the drones detect and track the victims before the arrival of human workers, increasing the chances of survival [2]. This only proves the validity of drones in this type of rescue missions.

Regarding the mechatronic integration of the launcher, it is possible to find some resemblances with other projects like the solar tracking systems used in photovoltaic

cells to obtain the maximum efficiency point. In this kind of projects, there is also a tower with different movements to point the cells, and these movements are similar to the ones that the launcher should be able to do. For these tracking systems, usually DC motors are used to control the linear and rotation movements [3].
Other previous works have been developed for this drone launcher. Some examples of them would be the development of a method to connect the drones via internet instead of radio or the design of the launcher, including its new launch method based on a sled. There is also a thesis where the design of the launcher is discussed, offering three different possibilities and finally choosing one. The location for the final installation is also considered, exposing positive and negative aspects of every location. Additionally, the materials needed for every piece and the mechanisms that can be used to make it move are debated too, and these conclusions will be used in this thesis to complete the mechatronic integration. [4]

1.2 Purpose and scope

The main purpose of this thesis is to design, build and test a working prototype of the launcher, including all the necessary mechanisms and sensors to provide the movements needed for a successful launch under different conditions.
The thesis will focus on the design of the different mechanisms necessary to perform a successful launch, which are pitch, lifting and rotation. After the design is completed and validated, the building of the prototype and the development of the control system are included as part of the project. Furthermore, the control system should be able to control the launcher from an external device, which includes the development of a Human Machine Interface (HMI) to command it in an easier way.

1.3 Ethical and Sustainability Issues

The aim of this thesis is the mechatronic integration of the drone launcher, however, the distinct parts involved in the project lead us to some ethical issues.
One of the examples of these ethical challenges is related with the usage of cameras, whether they are installed on the launcher or the drone itself. Privacy has been specially discussed in the last years because of the proliferation of different devices with embedded cameras, and how the information obtained from them can affect the privacy of the population. Including a camera should always come along with a report where it explains the usage of all the information acquired and the cause of its existence. In this case, it will help to detect the existence of any kind of danger around before the launching operation, such as persons or animals, but there should exist an indicator to show that the cameras are recording.
Another issue is related with the main function of the drone, which is helping in sea rescues. Since it involves people's lives, there is an important responsibility and not failure should be allowed. All the mechanisms should be trustworthy, so the experts can rely on them without caring about the conditions or the time elapsed since the last usage. Also, every system should be fast and easy to deploy because time is a very important factor in this type of rescues.

Furthermore, sustainability issues must be taken into consideration, because the drone launcher is intended to share spaces with humans, and also to work with them.
Linked with sharing spaces is the fact that the launcher must be installed outside, near the coast, which can have an impact on the environment. And not only because it can be dangerous, but also because it affects visually the landscape. The visual impact should not be underestimated and must be consequent with the surrounding area, trying to integrate it visually, as well as taking care of the sounds that might be emitted and the materials used.
The economical point of view is important too, especially in a society with no government funding. Although a fast return of the money is not expected in this project, the fact that it can improve the sea rescues by using the resources of the society in a better way can mean a reduction of the costs in future operations.
Finally, the social aspects of this project are directly linked with the quality of the sea rescues. If the drone manages to improve the reaction capacity of the Swedish Sea Rescue Society and to offer a better service for everyone on the sea, it would mean a step forward in this field.

1.4 Report structure

This thesis is divided in four parts. The first one, design, covers the conception of different mechanism to move the launcher, and also an analysis of its viability and component selection. The main focus of this part is to reflect the real design work, based on modifications and iterations to arrive to the final solution, and also expose the reasons behind these changes.
After that, in the control section, the control system developed is explained. Afterwards, the results of the real prototype are shown in the results part, and finally, in the conclusion, the disadvantages and the recommended improvements for further stages of the project are indicated.

2
Design

2.1 Movement requirements

2.1.1 Pitch

Depending on the wind force and direction, the launcher should be able to calculate the most convenient pitch angle to launch the drone. On the standby position, the capsule must remain perpendicular to the floor.

Regarding the speed, it should be higher than 3 °/s, meaning that it can move 90° in 30 seconds.

2.1.2 Rotation

As in the pitch movement, the launcher needs to rotate to point to the most appropriate direction depending on the wind. The movement needs to cover a full rotation of 360°, not only to face the launch direction, but also to protect the drone from external threats such as the sun or the wind. A movement where the launcher follows the position of a person when they look at it can also be studied for marketing purposes.

The necessary speed should be around 6 °/s so it can rotate completely in one minute.

2.1.3 Lift

Based on previous tests and prototypes, the capsule containing the drone should remain 1,2 m above the floor when it is in the standby position. When it is in this state, the drone can be shown to the people nearby and the launcher can also be used for advertisement.

During the launch, the capsule should remain a minimum of 2,4 m above the floor to assure that no one can reach the upper part, leading to a dangerous situation.

The elapsed time to extend the column should be less than a minute, meaning that the speed should be around 20 mm/s.

Finally, the column should be able to lift 1000 N and support a minimum momentum of 1000 Nm.

2.1.4 Speed consideration

For every different movement, the speed requirements have been specified, however, the final speed will depend on the slowest movement. Consequently, it is possible to choose slower movements that the listed if they move and finish at the same time as the slowest one. Nonetheless all the movements must be finished before one minute, because speed is a critical factor for the launcher to be useful.

2.2 Worm gearbox mechanism

2.2.1 Working principle

This mechanism is based on the usage of DC motor and gearboxes to provide the movement. Since the nominal torque of these motors is not very high, gearboxes need to be included to fulfil the requirements. This design was developed by a former master student, Filip Svalander.

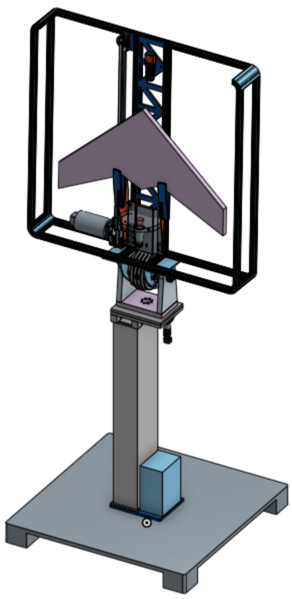
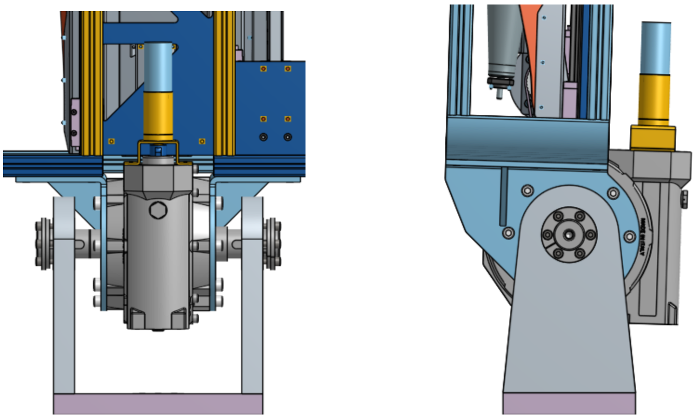


Figure 2.1: Isometric view of the worm gearbox launcher, designed with *Onshape*.

2.2.1.1 Pitch

The pitch movement is achieved with a worm gearbox, where the capsule is attached to it. The worm gearbox is powered by a DC motor attached to a gearbox on the back part. The horizontal axis is supported by two columns on both sides, with a bearing inside each column.

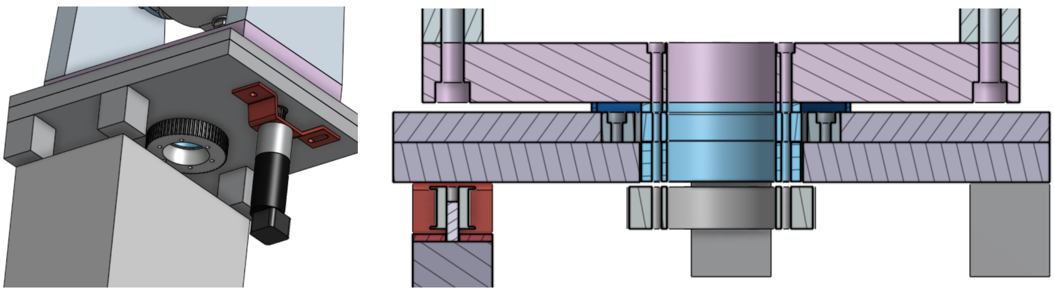


(a) Back view (b) Left view

Figure 2.2: Worm gearbox launcher pitch mechanism.

2.2.1.2 Rotation

The rotation is provided by another DC motor and a gearbox. The output of the gearbox is attached to a gear, connected to a bigger gear using a belt. This bigger gear is mounted on the base of a rotation platform linked with its upper part as well. In between these two platforms there is a bearing that makes possible the rotation movement. The inner part of the bearing is attached to the base and the upper platforms and the exterior is mounted on an intermediate platform.



(a) Isometric view (b) Section view

Figure 2.3: Worm gearbox launcher rotation mechanism.

2.2.1.3 Lift

A commercial lifting column can be used to lift the capsule. It will provide better results than designing a specific column for the prototype because it is a standard technology widely used in industrial environments.

2.2.1.4 Launch

This is the only part of the project that has been built and tested. It consists on a Siboni DC motor that transmits the movement to a platform using a belt. This platform carries the drone and slides along two rails to launch it. The platform is stopped using an elastic rope attached to the back part.

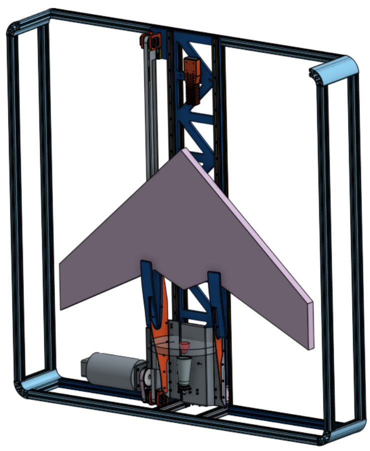


Figure 2.4: Launch capsule.

2.2.2 Component selection

2.2.2.1 Pitch motor and gearbox

For the calculation of the pitch motor, two assumptions will be made:

- Mass of the capsule ($m_{capsule}$) is 100 kg.
- Centre of mass (d_{mass}) is at 0,6 m.

These two assumptions are higher than the expected final result. The weight of the capsule is expected to be reduced to 70 kg and most of the weight is located on the back, meaning that the distance to the centre of mass would be shorter. The reason to calculate with these numbers is to add a safety factor to the components.

$$M_{required} = m_{capsule} \cdot d_{mass} \cdot g = 100 \cdot 0,6 \cdot 9,81 = 588,6\ Nm \tag{2.1}$$

The worm gearbox selected is Hydro-Mec Worm gearbox B 110 FB 53,00 C D -M V5.
Hydro-Mec Worm gearbox B 110 FB 53,00 C D -M V5 technical specifications:

- Output torque: $M_{worm\ o} = 620$ N.
- Gear ratio: $i_{worm} = 53$.

$$M_{worm\ i} = \frac{M_{worm\ o}}{i_{worm}} = \frac{620}{53} = 11,89\ Nm \tag{2.2}$$

The motors that have already been pre-selected are Maxon DC motors from the RE program. To not alter the previous design, a motor from that program will be selected. Most of the alternatives have similar specifications and prices, for that reason, the 24 V is selected. This motor is consistent with the rest of the components because they all operate under 24 V, which can help to simplify the installation not having to transform different voltages. The motor selected is Maxon RE40 148867.
Maxon RE40 148867 technical specifications:

- Nominal speed: $n_{motor} = 6940$ rpm.

- Nominal torque: M_{motor} = 0,177 Nm.

$$i_{pgearbox} = \frac{M_{worm\ i}}{M_{motor}} = \frac{11,89}{0,177} = 67,17 \tag{2.3}$$

The closest gearbox compatible with the motor is Maxon 203122.
Maxon 203122 technical specifications:

- Gear ratio: $i_{p\ gearbox}$ = 66.

Using this configuration, the final pitch specifications are:

$$M_{worm\ f} = M_{motor} \cdot i_{p\ gearbox} \cdot i_{worm} = 0,177 \cdot 66 \cdot 53 = 619,15\ Nm \tag{2.4}$$

The final torque is above the required torque (619,15 Nm > 588,6 Nm).
To determine if the speed is enough, the time required is:

$$V_{pitch} = \frac{n_{motor}}{i_{r\ gearbox} \cdot i_{worm}} = \frac{6940}{66 \cdot 53} = 1,98 rpm = 11,9\ °/s \tag{2.5}$$

$$T_{pitch\ 90} = \frac{90}{V_{pitch}} = \frac{90}{11,9} = 7,56\ s \tag{2.6}$$

The time required is acceptable for the operation of launching the drone.

2.2.2.2 Rotation motor and gearbox

As in the previous case, there are two assumptions used for the calculation:

- Mass of the capsule ($m_{capsule}$) is 100 kg.
- Measures of the capsule are 1,4 x 1,4 x 0,22 m.
- Rotation of the capsule is centred on the column.

For the same reasons as in the pitch calculation, a Maxon DC 24 V from the RE program is tested. Since there is only one 24 V motor in this program, the motor selected is the same that in the pitch movement (Maxon RE40 148867).
All the Maxon gearbox compatible with this motor have a maximum torque input of 15 Nm.

$$i_{rgearbox} = \frac{M_{maxgearbox}}{M_{motor}} = \frac{15}{0,177} = 84,745 \tag{2.7}$$

The closest gearbox compatible with the motor is Maxon 203124.
Maxon 203124 technical specifications:

- Gear ratio: $i_{r\ gearbox}$ = 81.

Based on the sizes of the gears from the rotation mechanism, its gear ratio is 5.
With all the gear ratios, the final torque and speed can be calculated:

$$M_f = M_{motor} \cdot i_{rgearbox} \cdot i_{beltgears} = 0,177 \cdot 81 \cdot 5 = 71,685\ Nm \tag{2.8}$$

$$V_{rotation} = \frac{n_{motor}}{i_{r\ gearbox} \cdot i_{belt\ gears}} = \frac{6940}{81 \cdot 5} = 17,136\ rpm = 1,794\ rad/s = 102,814\ °/s \tag{2.9}$$

To calculate the moment of inertia of the capsule that contains the drone, it can be approximated as a rectangular prism that rotates along its centre.

$$I_{capsule} = \frac{1}{12} \cdot 100 \cdot \left(h^2 + w^2\right) = \frac{100}{12} \cdot \left(1,4^2 + 0,22^2\right) = 16,736 \; kg \cdot m^2 \qquad (2.10)$$

Using the torque and the inertia, it is possible to calculate the angular acceleration and the time to reach the nominal speed:

$$M_f = I_{capsule} \cdot \alpha_{max} \qquad (2.11)$$

$$71,685 = 16,736 \cdot \alpha_{max}$$

$$\alpha_{max} = 4,283 \; rad/s^2$$

Which means that the nominal speed can be reached in:

$$T_{rotation} = \frac{1,794}{4,283} = 0,41 \; s \qquad (2.12)$$

It is not necessary to reach the acceleration of 4,283 rad/s^2, which means that the torque is higher than necessary. However, taking into account that no frictions have been calculated and the small difference in price by choosing a smaller motor, it is recommended to select the listed components.
The final speed of 102,814 °/s is too fast, but it can be controlled using a motor controller.

2.2.2.3 Lifting column

X2 Technology is a lifting column supplier that can provide a commercial column inside the specifications. Among their catalogue, the most appropriate option for the launcher is the DB model. However, the height requirements are not standard, and they have to be personalised for this application. It is powered using 24 V, it requires 6 A and it can lift 2000 N.

2.2.3 Motor controllers

2.2.3.1 Current necessary to launch the motor

The motor that is going to be used for launching the drone is already selected and installed. For that reason, it is important to know how much current it will need during its operation in order to select the motor controller that fulfils the requirements.
The calculation will be based on four assumptions:

- Drone weight is 0,5 kg.
- Sled weight is 1,5 kg.
- Final launch speed (V_f) is 10,5 m/s.

- Launch distance (d_{launch}) is 1,1. The total length is 1,37 m but only 1,1 m are useful due the size of the sled.

First, it is necessary to calculate the acceleration to launch the drone, based on the final speed and the distance available:

$$V_f^2 - V_0^2 = 2 \cdot a \cdot d_{launch} \tag{2.13}$$

$$10,5^2 - 0 = 2 \cdot a \cdot 1,1$$

$$a = 50,1136 \ m/s^2$$

Then, it is possible to estimate the time, force and work necessary using the final speed, the mass and the distance:

$$T_{launch} = \frac{V_f}{a} = \frac{10,5}{50,1136} = 0,21 \ s \tag{2.14}$$

$$F_{launch} = m_{drone \ sled} \cdot a = 2 \cdot 50,1136 = 100,22 \ N \tag{2.15}$$

$$W_{launch} = F_{launch} \cdot d_{launch} = 100,22 \cdot 1,1 = 110,24 \ J \tag{2.16}$$

The power can be measured

$$P_{launch} = \frac{W_{launch}}{T_{launch}} = \frac{110,24}{0,21} = 525 \ W \tag{2.17}$$

And finally, the current can be defined using Ohm's law, knowing that the motor is powered with 24 V.

$$I_{launch} = \frac{P_{launch}}{V_{launch}} = \frac{525}{24} = 21,875 \ A \tag{2.18}$$

2.2.3.2 Commercial motor controllers

Motor controllers can have big differences in their price depending on the current that they can handle. In order to reduce the final budget of the prototype, it is recommended to use the smaller controller that can drive the necessary current. Also, having current sensing has been established as a requirement because it can lead to a better control and reduce the accidents caused by an incorrect power supply. Finally, it is advisable to choose double channel motor controllers if it is possible, because it reduces the price of the individual ones as they share some of their components. To do that, both motor should have similar specifications.

2.2.3.2.1 Rotation and pitch motors

Since they are both the same motor, the selected motor controller is a dual channel controller. The current supported is higher than the nominal current, but the price difference with its smaller version is small enough to not consider it. The selected controller is RoboClaw 2x15A, that can support 24 V and 15 A per channel (30 A peak).

2.2.3.2.2 Launch motor and lifting column

For the launch motor, the current calculated is below the nominal current of the motor (48 A). However, the efficiency and the friction have not been included, so choosing a motor controller that can handle more than the calculated current is advisable.

The lifting column nominal current is 6 A, but from an economic point of view, it is cheaper to use the same double channel controller that the one used on the launch motor. The model selected, RoboClaw 2x30A, is produced by the same supplier as the previous motor controller, making the control easier. This model is able to support 30 A and 60 A peak.

2.2.4 Conclusions of the mechanism

The mechanism is valid and most of the components are ready to be tested on a prototype. Nevertheless, it has some flaws, for example, all the torque needs to be supported by the worm gearbox, which leads to choose high quality components, increasing the final price. Besides, the pitch motor has to be installed on the back part of the capsule, making the design less visual appealing and having critical components more exposed.

The company agreed to look for an alternative solution to compare it and decide the best option for the prototype.

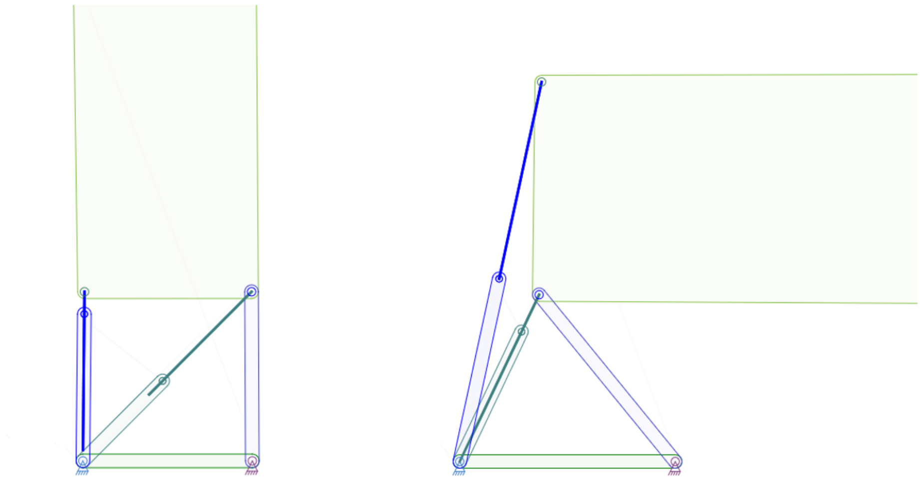
2.3 Alternative mechanisms

2.3.1 Two independent linear actuators

The main purpose of this design was to have a flat base, so the previous rotation mechanism could still be used. Moreover, by having two linear actuators the loads are not as high in each one of them, reducing their size to have a more compact design.

The main drawback of this mechanism is that it is harder to implement because both actuators must be coordinated. Depending of the relative movement there are infinite positions but not all of them are stable or secure enough. Finally, the launch position is not well balanced and it is not very reliable.

Despite all the negative aspects, the idea of using linear actuators seemed appropriate for the project since they are simple and able to supply a big force using little space. Based on that, more designs were developed using the same principle.

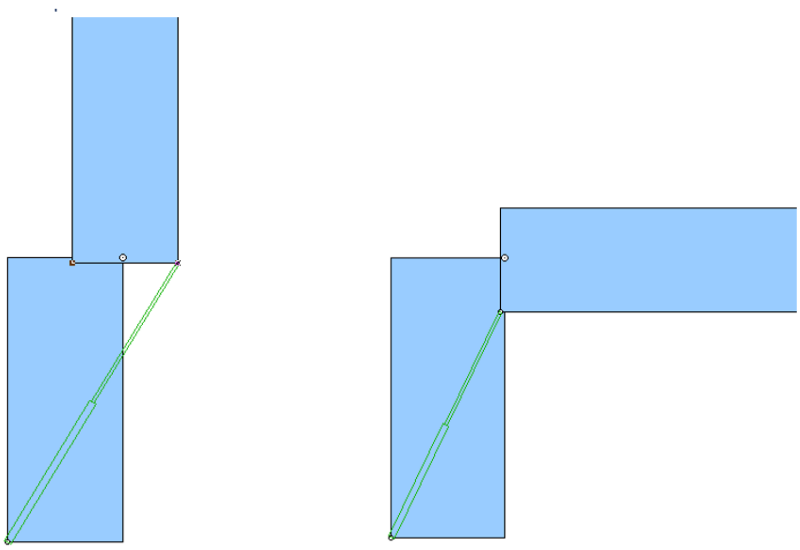


(a) Standby position (b) Launch position

Figure 2.5: Two independent linear actuators mechanism designed using *Linkage*.

2.3.2 Two parallel linear actuators

In this case, the maximum simplicity was sought. The design is based on having a rotation axis in the middle of the capsule and two linear actuators attached to two of its corners. Additionally, the capsule could lay on a surface when it is on standby position to make it more stable.



(a) Standby position (b) Launch position

Figure 2.6: Two parallel linear actuators mechanism designed using *Working Model*.

Despite its simplicity, a deeper study of the components showed that it is very inefficient in terms of force. Using the software *Working Model* to perform a simulation proved that a force of 17000 N was needed to rotate the capsule if it weighted 100 kg, and 12000 N for 70 kg. The distance between the point where the force is applied and the rotation axis is very short, in consequence, the momentum created needs a big force to start the movement. And this distance cannot be extended because the width of the column.

In addition, commercial linear actuators that can provide this load are very long, forcing the mechanism to be very tall. This is very problematic because, if it is mounted on the top of a lifting column, the retracted length would be too short to have the necessary stroke. Finally, the capsule is not centred, and the actuators are exposed.

2.4 Vertical linear actuator mechanism. First iteration

2.4.1 Working principle

To solve the drawbacks of the previous designs, a mechanism with a vertical linear actuator and a slider was designed.

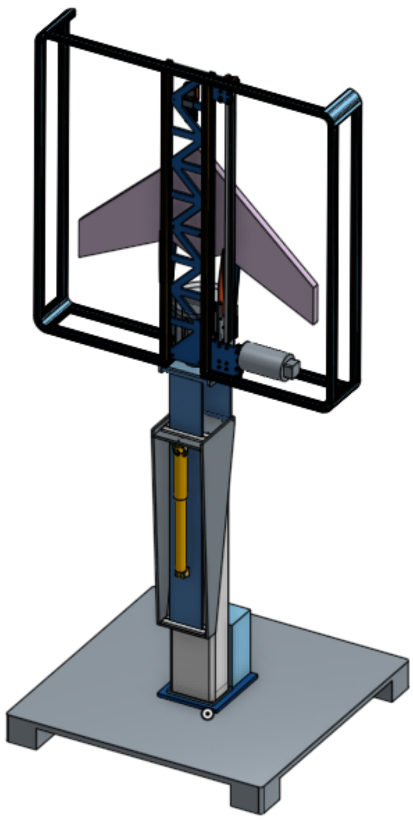


Figure 2.7: Isometric view of the vertical linear actuator launcher, designed with *Onshape*.

2.4.1.1 Pitch

The linear actuator is attached with brackets to the upper plate of the column and to a vertical slider. When the actuator is extended, the slider moves down, pushing the capsule to be inclined up to 90°. The sliders glides between two wholes covered by high-density polyethylene to reduce the friction.

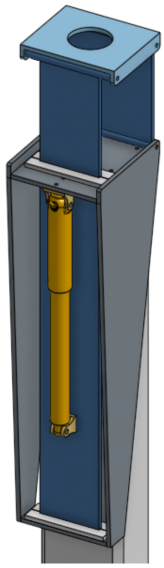


Figure 2.8: Isometric view of the pitch mechanism.

2.4.1.2 Rotation

The rotation cannot be done from the top of the column because of the new pitch mechanism, so it must be done from the base. At this point of the project, the rotation mechanism was not designed for this iteration.

2.4.1.3 Lift

The lifting mechanism is exactly like the one used in the worm gearbox mechanism, using the same lifting column.

2.4.2 Design validation

2.4.2.1 Necessary force for the linear actuator

To calculate the force that the linear actuator should provide to lift the capsule the software *Working Model* has been used. The measures have been obtained from the assembly so some differences with a real prototype should be expected. Despite that, the numbers offered by the simulation should be used to determine the magnitude order, testing if the solution is viable or not.

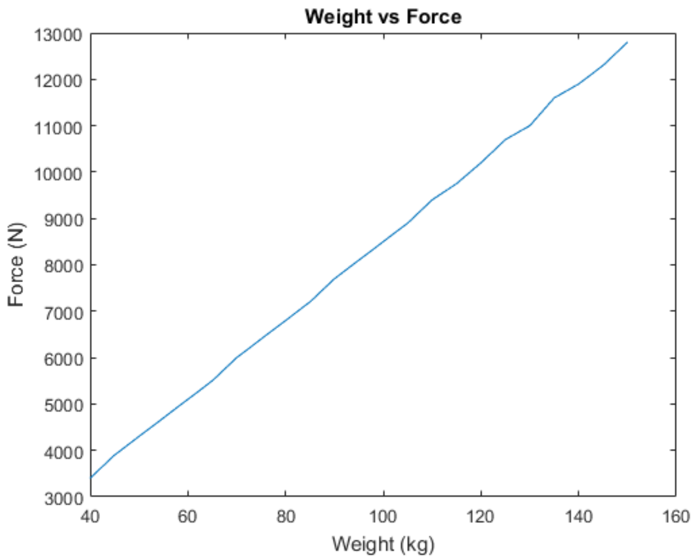


Figure 2.9: Comparison between the necessary force of the linear actuator depending on the weight of the capsule.

The correlation between weight and force is completely linear and it can be translated directly into an equation:

$$Force = 84,96 \cdot Weight + 22,233 \tag{2.19}$$

At this stage of the project, the definitive weight of the capsule has not been defined yet. Following the simulation, it would be advisable to reduce the weight of the capsule as much as possible to be able reduce the actuator force and the loads on the materials.

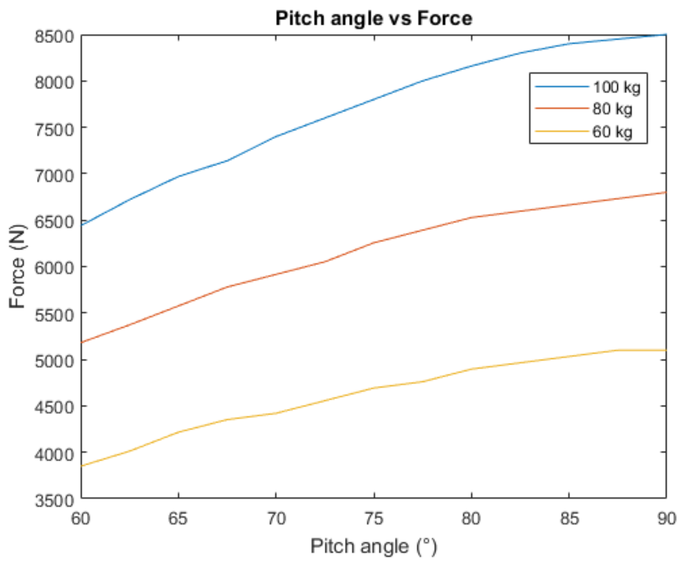


Figure 2.10: Comparison of the necessary force of the linear actuator depending on the pitch angle, for a capsule weight of 100 kg, 80 kg, and 60 kg.

The simulation shows that restricting the pitch angle to a value below 90° does not translate into much lower forces from the linear actuator. The slope is expected to be steeper below 60°, but that angles is too restrictive and useless for the project. The behaviour is almost the same after changing the weight of the capsule.
To sum up, it is not recommended to reduce the pitch angle below 90° because the loss in functionality from the launcher only reduces the necessary force from the actuator in a small percentage.
For the following calculations, a necessary force of 6000 N will be used, because it has been considered enough to test the prototype and it is easy to find a commercial actuator with cylindrical shape, which is helpful to reduce the width of the column.

2.4.2.2 Wind load

The highest wind load will appear when the launcher is in the standby position, because the capsule is perpendicular to the wind.
The following assumptions have been made:

- Drag coefficient (C_d) is 1,28 because it is a flat surface [5].
- The air density (ρ_{wind}) is 1,2 kg/m^3 [6].
- The maximum wind speed that the launcher must support (V_{wind}) is 25 m/s.

The wind load on the capsule is calculated using the drag equation [7]:

$$F_{wind} = \frac{1}{2} \cdot C_d \cdot \rho_{wind} \cdot A_{capsule} \cdot V_{wind}^2 = \frac{1}{2} \cdot 1,28 \cdot 1,2 \cdot 1,96 \cdot 25^2 = 918,75\ N \quad (2.20)$$

Nevertheless, the wind load could be reduced making use of the rotation movement in next iterations of the project. By using wind sensors, if the wind is too strong the column could rotate to avoid the wind to arrive perpendicularly to the surface. Also, the wind speed of 25 m/s is very unusual in Gothenburg, where, during the windiest day of the year, 90 % of the wind is slower than 7,6 m/s. [8]

2.4.2.3 Maximum momentum on the top of the column

The momentum on the top of the column caused by the wind load can be approximated as:

$$M_{wind} = F_{wind} \cdot d_{wind} = 918,75 \cdot 0,95 = 872,81\ Nm \quad (2.21)$$

From the assembly, the distance measured between the centre of the column and the point where the force of the linear actuator is applied is 0,157 m. This vertical force creates a momentum that must be supported by the column and it is calculated with:

$$M_{actuator} = F_{actuator} \cdot d_{actuator} = 6000 \cdot 0,157 = 942\ Nm \quad (2.22)$$

The weight of the capsule itself also creates a momentum in the opposite direction, but it has not been considered to increase the safety factor.
Both momentums are similar, meaning that the minimum momentum that the column must support should be higher than 1000 Nm.

2.4.2.4 Horizontal load

Assuming that the weight of the capsule is 100 kg, that the centre of mass is located at a distance of 0,7 m and the width is 0,22 m. If the weight is directly transformed into a momentum on the edge, the horizontal forces on the corners are calculated as:

$$F_{horizontal} = \frac{m_{capsule} \cdot 9,81 \cdot d_{mass}}{w} = \frac{100 \cdot 9,81 \cdot 0,7}{0,22} = 3121,36 \ N \qquad (2.23)$$

2.4.2.5 Support loads due to the momentum on the bracket

Considering the distances offered by the brackets suppliers and a force of 6000 N, it is necessary to calculate the momentum at the bracket.

$$M_{bracket} = F_{actuator} \cdot d_{bracket} = 6000 \cdot 0,042 = 252 \ Nm \qquad (2.24)$$

This momentum is directly applied to the slider and it must be supported by the horizontal constraints, which in this case are mode out of polyethylene. The constraints are approximated as two roller supports where the axial forces are not considered because they are responsible of the movement.

$$\Sigma F = 0 \rightarrow R_1 = R_2 \qquad (2.25)$$

$$R \cdot 0,842 = 252 \qquad (2.26)$$

$$R_1 = R_2 = 299,29 \ N$$

Compared with the horizontal load of the capsule, it is much lower, so it can be assumed that the worst scenario appears when the pitch angle of the capsule is 90°.

2.4.2.6 Friction and compression

Assuming the horizontal force calculated in equation 2.23 is directly applied to the polyethylene and that the coefficient of friction is 0,15 [9].

$$F_{friction} = \mu \cdot N = 0,15 \cdot 3121,36 = 468,204 \ N \qquad (2.27)$$

The compressive strength of the polyethylene is 28 MPa [10]. If the width of the slider is considered as 180 mm and the width of the polyethylene is 25 mm.

$$P_{PE} = \frac{F_{horizontal}}{A_{PE}} = \frac{3121,36}{0,18 \cdot 0,025} = 0,832MPa = 8,487 \ kg/cm^2 \qquad (2.28)$$

Which is far from the compressive strength of the polyethylene, so it will not fail under compression, but it can be affected by wear. Although the numbers seem reasonable, they are very approximated. Assuming a friction of 500 N, the sliding mechanism is not invalidated, and an extra actuator could be installed on the prototype if more force is needed.

2.4.2.7 Bending of the slider due the horizontal load

The material chosen for all the mechanised parts is aluminium because it has a low weight and it is easy to mechanise. The horizontal load in the slider when the pitch angle is 90° is the worst scenario for the bending. The calculation is based on the following assumptions:

- Force ($F_{horizontal}$) is 3121,36 N, as calculated in equation 2.23.
- Distance between load and the support (L) is 10 cm.
- Based on a beam with a fixed support on one end and free on the other one.
- Young's modulus of aluminium (E) is 69 GPa [11].
- Width of the slider (w_{slider}) is 150 mm.

The area moment of inertia inertia of the slider is calculated as [12]:

$$I_{slider} = \frac{1}{12} \cdot w_{slider} \cdot thickness^3 = \frac{1}{12} \cdot 0,15 \cdot thickness^3 \tag{2.29}$$

The displacement of the free ending is determined as [12]:

$$y = \frac{F_{horizontal} \cdot L^3}{3 \cdot E \cdot I_{slider}} = \frac{3121,36 \cdot 0,1^3}{3 \cdot 69 \cdot 10^9 \cdot \frac{0,15}{12} \cdot thickness^3} \tag{2.30}$$

Depending on the thickness of the slider, the displacement is:

$$thickness = \left\{ \begin{array}{l} 15mm \rightarrow y = 0,357\ mm \\ 10mm \rightarrow y = 1,206\ mm \\ 5mm \rightarrow y = 9,650\ mm \end{array} \right\}$$

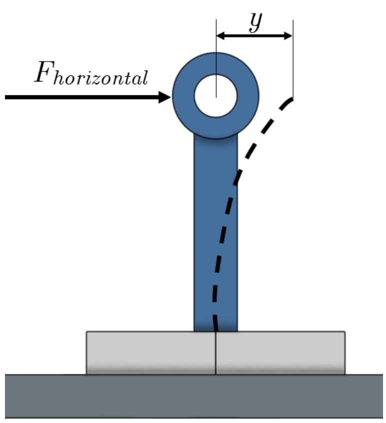


Figure 2.11: Bending of the slider due the horizontal load.

It is not recommended to have a thickness below 10 mm because the bending would be excessive. The force is directly proportional to the displacement, so even if the force is twice the considered, the displacement would still be acceptable. Based on that reasons, a thickness of 10 mm is chosen for the prototype, as it considered thick enough to support the horizontal load and to attach the bracket.

2.4.2.8 Buckling of the slider due the axial load

To restore the standby position after the launch, the slider must support and axial load of 6000 N that can make it buckle. By calculating the critical load, it is possible to determine when the slider reaches the unstable equilibrium that would make it fail by buckling, but some displacement can still occur. The Euler's formula is used to establish the critical load, considering some assumptions:

- Only the axial load is considered, but there are other forces and momentum that can decrease the critical load.
- Thickness of the slider is 10 mm, width is 150 mm and length is 0,95 m.
- Intermediate support is not considered to increase safety factor.
- One end is considered fixed and the other one articulated ($\beta = 0{,}7$).

As in equation 2.29, the area moment of inertia of the slider is [12]:

$$I_{slider} = \frac{1}{12} \cdot w_{slider} \cdot thickness^3 = \frac{1}{12} \cdot 0,15 \cdot 0,01^3 = 12,5 \cdot 10^{-9}\ m^4 \tag{2.31}$$

Euler's formula for the critical load [13]:

$$P_{cr} = \frac{\pi^2 \cdot E \cdot I_{slider}}{(\beta \cdot L)^2} = \frac{\pi^2 \cdot 69 \cdot 10^9 \cdot 12,5 \cdot 10^{-9}}{(0,7 \cdot 0,95)^2} = 19249,327\ N \tag{2.32}$$

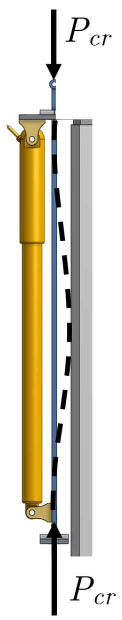


Figure 2.12: Buckling of the slider due the axial load.

The critical load is much higher than the load considered of 6000 N, so the slider will not fail by buckling. However, it is possible that it exists a non-critical displacement in the middle of the slider, but there is enough perpendicular space not to be a problem.

2.5 Vertical linear actuator mechanism. Second iteration

To avoid all the drawbacks of the first iteration, a new design is developed using the same working principle.

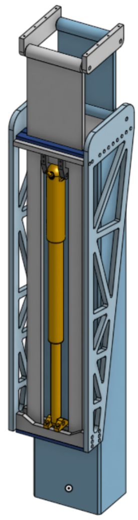
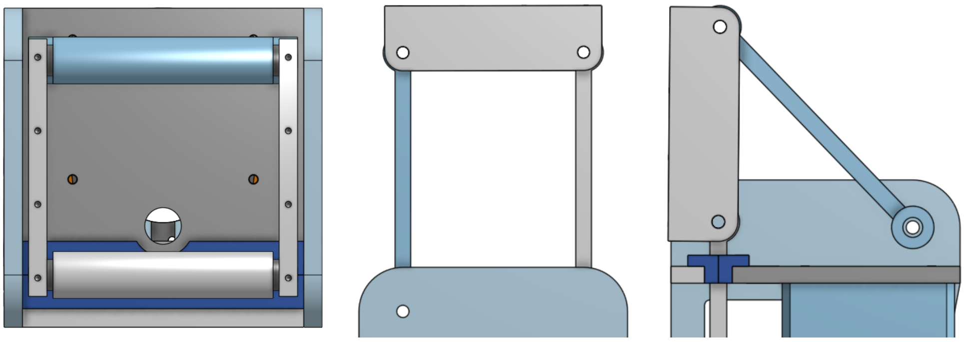


Figure 2.13: Isometric view of the second iteration of the vertical linear actuator mechanism.

The main changes are:

- A pattern of holes is introduced on the side plates to reduce its weight but maintaining its strength.
- Increased the distance between the side plates. Now the capsule support can fit inside them, making it more stable and easier to reach 90°. Moreover, the horizontal load still exists but the bending caused by them is almost negligible because the distance between the load and the support has been drastically reduced, as it can be seen in figure 2.14 (c).
- Augmented the length of the capsule support. By doing that, the distance between the slider and the other column also increases, reducing in more than a 30 % the forces necessary for the pitch.
- The load from the linear actuator is now almost directly applied as an axial force in the column, instead of creating a momentum.
- Thickness has been changed to 15 mm to increase the safety factor and for aesthetic reasons. Since it is the only part of the mechanism that is not going to be covered, it looks more robust. Also, it is more convenient for screwing all the components together.
- The mechanism is now centred in the middle of the column as figure 2.14 (b) shows. The measures of the components have been adjusted as well to have a square shape from the top view, which can be seen in figure 2.14 (a).
- The linear actuator is now integrated in the slider by having a whole in it, reducing the total width of the mechanism and helping to apply the force

directly to the centre of the slider. By doing that, the length of the slider that has to support the axial load is reduced, and the risk of buckling is lower.
- All the metallic components have the same thickness, so they can be mechanised from the same sheet of metal.

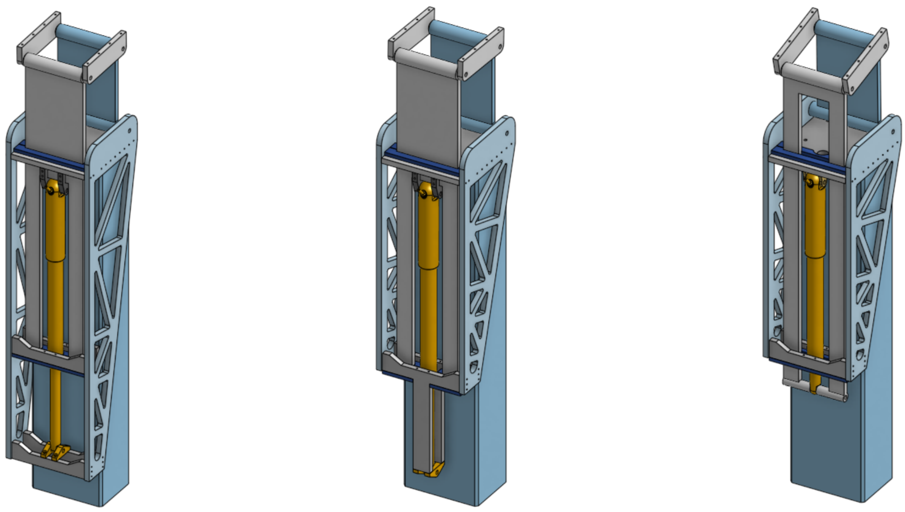


(a) Top view (b) Left view (c) Right view inclined

Figure 2.14: Different views of the second iteration.

2.5.1 Column length issue

However, the necessary column length to be able to reach 90 ° with the pitch mechanism is 1,2 m. This length, plus the length of the slider and column when it is in the standby position, would place the capsule 40 cm higher than expected. Having the capsule that high would make more difficult to have the drone exposed in a good eyesight.
In consequence, it is necessary redesign the pitch mechanism to reduce its height. First, several alternatives were modelled, based on integrating the linear actuator between the supports of the slider.



(a) Shortened slider (b) "T" support (c) Gap in the slider

Figure 2.15: Alternatives to reduce height in the pitch mechanism.

2.6 Vertical linear actuator mechanism. Third iteration.

Finally, another iteration is developed with reduced height.

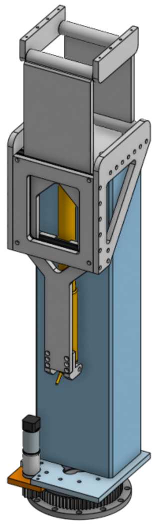


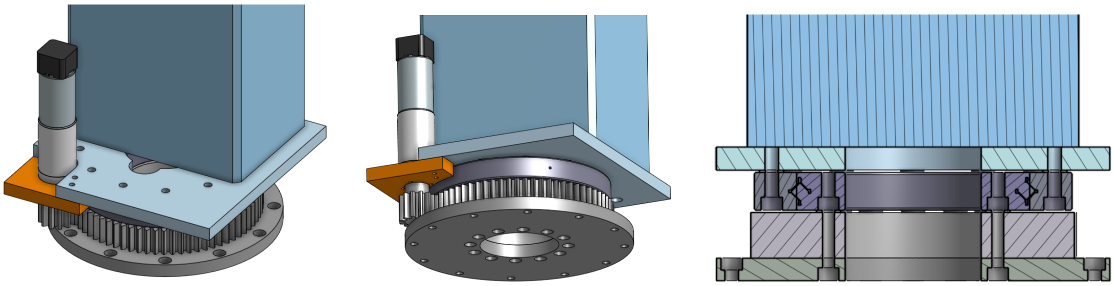
Figure 2.16: Isometric view of the third iteration of the vertical linear actuator mechanism.

The main differences with the previous iteration are:

- Linear actuator displaced to the back part of the slider. Now it is attached between the upper plate and the bottom of the slider.
- Reduced the distance between the supports of the slider, making the side plates smaller and much simpler.
- Increased again the distance between the slider and the column.
- Increased the sliding surface with the polyethylene. Now it covers the vertical distance of the slider instead of only sliding in the horizontal surfaces.
- Reduced the width of the slider in the bottom part. By doing this, it is possible to have more space to install other components inside the column, such as batteries and the control devices.
- Reduced the weight of the slider with a central hole.
- Added rotation mechanism.

2.6.1 New rotation mechanism.

The rotation mechanism has to rotate the column from the base. It is based on the same principle as in section 2.2.1.2, the column is mounted on the outer part of a bearing while the inner part is fixed to the ground. A gear is also connected with the inner part, below it.
The rotation movement is achieved with a small gear that turns around the fixed bigger gear. The force is provided by a DC motor, whose model is the same as the one calculated in section 2.2.2.2, Maxon RE 40 148867 and gearbox Maxon GP 42 C 203124.



(a) Top isometric view (b) Down isometric view (c) Section view

Figure 2.17: New rotation mechanism.

2.6.1.1 Bearing calculation

In order to be able to calculate the bearing, it is necessary to make the following assumptions:

- Axial load (F_a) is 1471,5 N assuming that the capsule weights 100 kg and the column weights 50 kg.
- Radial load (F_r) is 0 because the mechanism uses gears without belt.
- Momentum in the bearing ($M_{bearing}$) is 1837500 Nmm assuming that the maximum force is the wind load calculated in equation 2.20 and the distance between the centre of mass of the capsule and the base is 2 m, taken from the assembly.
- The commercial bearing selected for the calculations is Hiwin CRBE09025. This bearing is the biggest possible taking into account the size limit of the bottom plate.
- For that commercial bearing, the medium distance between the inner and outer ring (D_{pw}) is 150 mm, the dynamic load (C) is 46 kN and the static load (C_0) is 80,2 kN.

First it is necessary to calculate the radial and axial load coefficients: [14]

$$\frac{F_a}{F_r + \frac{2 \cdot M_{bearing}}{D_{pw}}} = \frac{1471,5}{0 + \frac{2 \cdot 1837500}{15}} = 0,06 \tag{2.33}$$

Since the value is below 1,5, the value of X is 1 and the value of Y is 0,45.
The radial load, axial load and momentum can be combined into a single load, called dynamic equivalent load, by using the next equation: [14]

$$P = X \cdot (F_r + \frac{2 \cdot M_{bearing}}{D_{pw}}) + Y \cdot F_a = 1 \cdot (0 + \frac{2 \cdot 1837500}{150}) + 0,48 \cdot 1471,5 = 25162,175\ N \tag{2.34}$$

As the dynamic equivalent load, the static equivalent load is determined with the following equation: [14]

$$P_0 = F_r + \frac{2 \cdot M_{bearing}}{D_{pw}} + 0,44 \cdot F_a = 0 + \frac{2 \cdot 1837500}{150} + 0,44 \cdot 1471,5 = 25147,46\ N \tag{2.35}$$

The basic life rating is used to calculate the revolutions that 90 % of the bearings will support after operating under the same operating condition. Is is calculated as: [14]

$$L_{bearing} = (\frac{C}{P})^{\frac{10}{3}} \tag{2.36}$$

The safety factor is: [14]

$$F_s = \frac{C_0}{P_0} = \frac{80,2}{25,147} = 3,189 \tag{2.37}$$

The suggested safety factor for high rotation and high accuracy according to the supplier is 3 [14], which means that the bearing is appropriate.

It would also be interesting to know the maximum force that the bearing can support applied as a horizontal force in the capsule. It can be obtained as:

$$C_0 = F_r + \frac{2 \cdot M_{max\ bearing}}{D_{pw}} + 0,44 \cdot F_a \tag{2.38}$$

$$80,2 = 0 + \frac{2 \cdot M_{max\ bearing}}{150} + 0,44 \cdot 14751,5$$

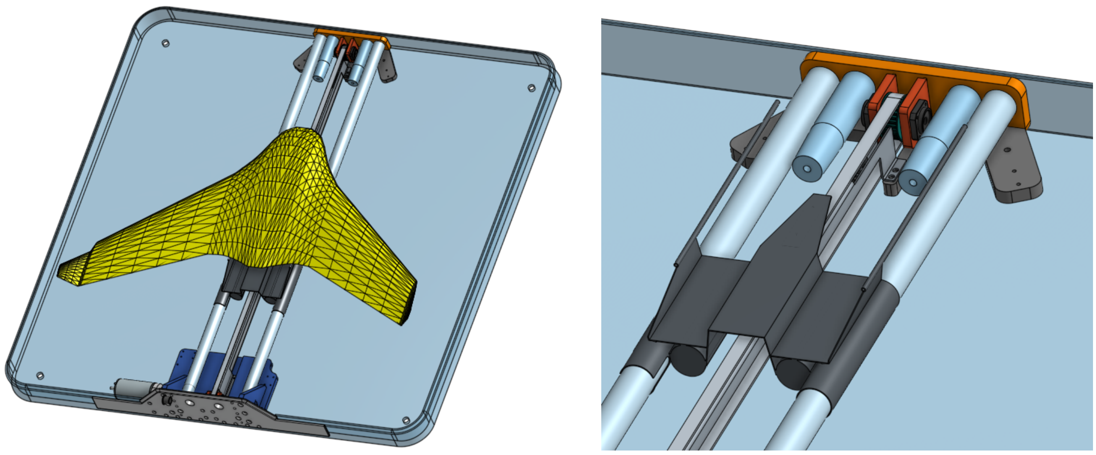
$$M_{max\ bearing} = 5966,44\ Nm$$

Which means that the maximum force admissible by the bearing in the capsule is 2983,22 N. This force is considered admissible.

2.6.2 New launch mechanism

Despite having it built, the launch capsule presents some drawbacks, sometimes the platform gets stuck because the movement is not completely parallel to the rails. Also the breaking system consisting in a elastic rope is not very refined and the platform is too heavy, which increases the necessary force of the motor.

Trying to use as much previous components as possible, a new mechanism is designed. The platform is now made out of carbon fibber to reduce the weight, and it brakes automatically thanks to the compressed air. There are two columns located at the last part of the capsule and they are introduced in the platform when it reaches the end of the capsule.



(a) Isometric view of the launch capsule (b) Detailed view of the launch mechanism

Figure 2.18: New launch capsule.

2.6.2.1 New launch motor

However, the existing launch motor is too big to be installed in this new design, and it is also too heavy, adding too much weight in a critical part. Furthermore, with a smaller motor the second belt would not be necessary and it would not block the drone when it is retracted. Based on the fact that the launch motor will be used during very short periods of time, it is possible to select a smaller motor and make it work above its nominal values for a short period of time.
Based on the following assumptions:

- Final launch speed (V_f) is 10,5 m/s.
- Radius of the wheel driving the belt ($r_{\text{belt launch}}$) is 20 mm.
- Launch time (T_{launch}) is 0,21 s (Equation 2.14).

The final angular speed of the motor should be:

$$V_f = r_{belt\ launch} \cdot \omega_{launch} \tag{2.39}$$
$$10,5 = 0,025 \cdot \omega_{launch}$$
$$\omega_{launch} = 420\ rad/s = 4010,7\ rpm$$

The motor Maxon 388984 is the one selected to prove the calculations.
Maxon 388984 technical specifications:

- Speed constant = 208 rpm/V.
- Speed / torque gradient = 0,275 rpm/mNm.
- Torque constant = 46 mNm/A.

The angular speed can be calculated using the speed constant and knowing that it will be powered with 24 V:

$$\omega_{launch} = 24 \cdot 208 = 4992\ rpm \tag{2.40}$$

Under a load of 2800 mNm, which is much more than the torque that we need, the speed will be limited by the gradient:

$$\omega_{launchf} = 4992 - (2800 \cdot 0,275) = 4222\ rpm \tag{2.41}$$

The current necessary is obtained using the torque constant and the torque:

$$I_{launch} = 2800/46 = 60,87\ A \tag{2.42}$$

Which means 6 times the nominal current. Besides, the motor controller selected can support such current. This means that under heavy load the motor is able to reach the speed with the equipment already available.

After contacting the supplier and showing these calculations, it confirmed that, although there are no guaranties, it should work safely since the launch time is too short to overheat the motor. The supplier also mentioned that the motor is able to reach that speed in 0,2 seconds.

2.6.3 Installation base and space for electrical components

A new base based on a circular design has been added to improve the stability of the column. As mentioned before, the width of the slider has been reduced in the bottom part to allow the installation of the batteries and the other electrical components. The size of this space is 100 x 70 x 330 mm and it can be seen, in red, in figure 2.19

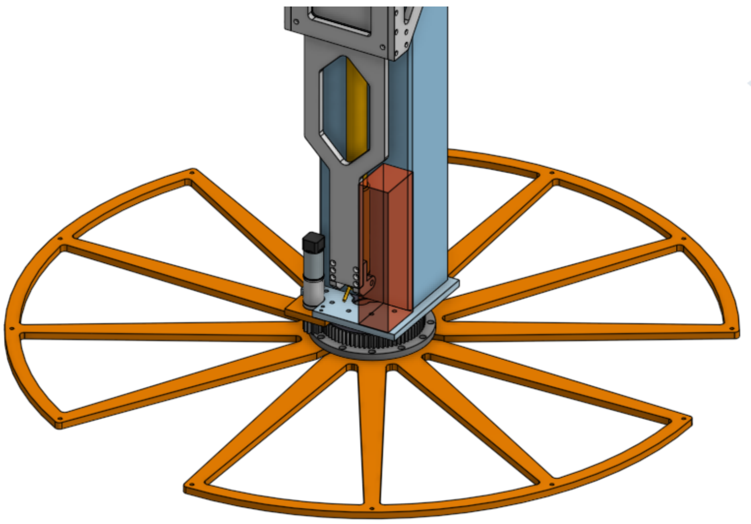


Figure 2.19: Installation base and space for electronic components.

2.6.4 New necessary force for the linear actuator.

Thanks to the increased distance between the slider and the column, the force has been greatly reduced. The results of a simulation using *Working Model* are shown in Figure 2.20.

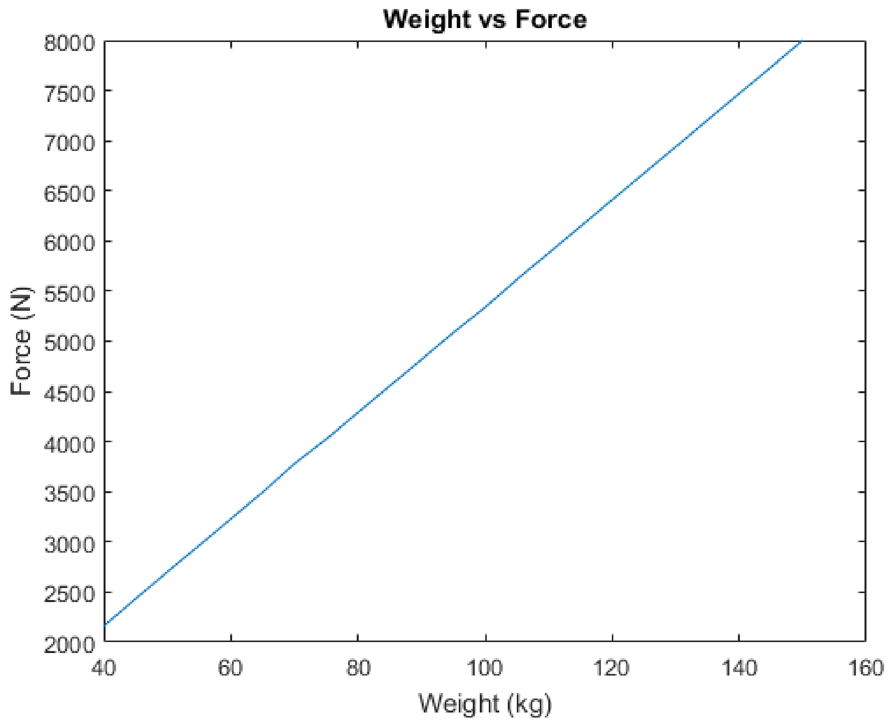


Figure 2.20: Comparison between the new necessary force of the linear actuator depending on the weight of the capsule.

In the previous iteration it was necessary a force of 8500 N for a capsule weight of 100 kg, while with the new design, a force of 5350 N is enough.
The force can be easily calculated following the equation:

$$Force = 53 \cdot Weight + 50 \tag{2.43}$$

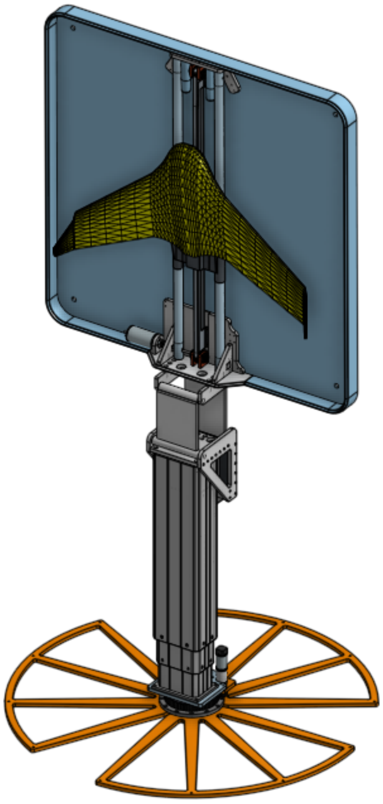
This equation can be useful to determine the force even if the pitch mechanism is changed, as long as the measures of the supports are maintained.

2.6.5 Validity of the previous calculations

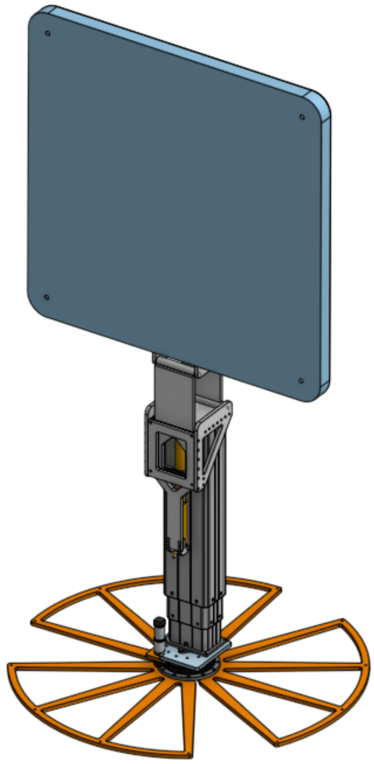
The intention of the previous calculations was not to obtain precise numbers about the mechanism, but to validate if the design was viable or not. For that reason, they are not calculated again, as all the loads are lower with the new design.
The only calculations that have changed are the momentum of the bracket, because the force is directly applied to their centre, and the bending and buckling of the slider, being lower than before.

2.6.6 Final assembly

The final CAD model includes all the changes performed after the iterations. It also includes the real assembly of the personalised lifting column.



(a) Front part



(b) Back part

Figure 2.21: Final assembly.

2.6.7 Selection of final components

2.6.7.1 Linear actuator

The linear actuator selected is ANT-52 from Sito Motor. One of the reasons to choose it is that it has cylindrical shape. Usually, high load linear actuators have a DC motor installed next to it, making them shorter but wider. In this project, the length is not a limitation but reducing width is essential to get a square shape for the column. Furthermore, the voltage is coherent with the rest of the components. The position control is achieved with a Hall sensor integrated in the actuator.

Figure 2.22: ANT-52 linear actuator from Sito Motor.

ANT-52 technical specifications:

- Voltage: 24 V.
- Current at maximum load: 6A.
- Load: 6000 N.
- Load speed: 3 mm/s.
- Stroke: 250 mm.
- Ingress protection: IP67.
- Minimum length: 575 mm.
- Maximum length: 825 mm.
- Hall sensor.

2.6.7.2 Rotation motor

The motor selected is the same as calculated in section 2.2.2.2 because the assumptions for the previous design are still considered valid. The motor is a brushed Maxon RE 40 148867 with the gearbox 203124 and an encoder.

Figure 2.23: Maxon DC motor RE 40 148867.

Maxon motor, gearbox and encoder technical specifications:

- Nominal voltage: 24 V.
- Nominal speed: 6940 rpm.
- Nominal torque: 177 mNm.
- Nominal current: 6 A.
- Gearbox gear ratio: 81.
- Encoder counts per turn: 500.

2.6.7.3 Launch motor

The motor selected is the one chosen in section 2.6.2.1, a brushed Maxon RE 65 388984 with an encoder. The nominal voltage is 18V, but considering the small time required to launch the drone, it is acceptable to power it using 24 V. Physically it is very similar to figure 2.23.
Maxon motor, gearbox and encoder technical specifications:

- Nominal voltage: 18V.
- Nominal speed: 3250 rpm.
- Nominal torque: 412 mNm.
- Nominal current: 10 A.
- Encoder counts per turn: 500.

2.6.7.4 Lifting column

The same column as the one chosen in section 2.2.2.3 is selected, because the requirements of 100 kg for the axial load and 1000 Nm for the minimum momentum on the top have not changed. However, after a meeting with the supplier, some design changes were advised:

- If the column is mounted upside down, it is possible to reduce the problems cause by the condensation. By mounting it like that, the water can be expelled instead of getting stuck inside.
- It is possible to mount the column on a base to increase the stiffness, reducing the momentum that the bolts have to support. This base can only be installed if the column is mounted upside down.

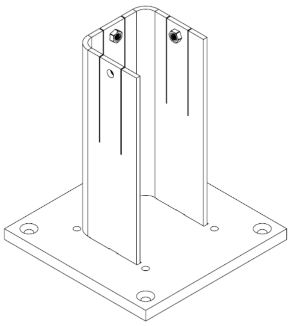


Figure 2.24: Lifting column base.

- Three sections are needed to reach the length specification, but given the length of the column, it would be a good idea to increase the overlap between sections for safety reasons. The equation that they use to define the stroke is:

$$Stroke = (Retracted\ length - 300) \cdot 2 \tag{2.44}$$

By using equation 2.44, the ideal column would have a retracted length of 0,9 m and a stroke of 1,2 m.

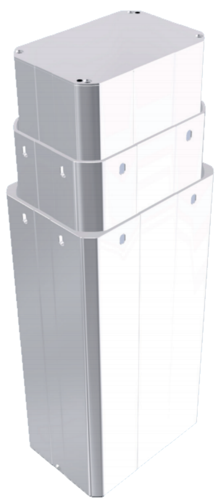


Figure 2.25: Lifting column X2 model DB.

Lifting column technical specifications:

- Force: 2000 N.
- Voltage: 24 V.
- Current: 6 A.
- Speed: 25-13 mm/s

2.6.7.5 Rotation gear wheels

For the rotation movement, two gear wheels are needed. The selected gears are M3x70-P and M3x12, whose module and sizes were chosen in order to maximise the gear ratio depending on the available space under the column and the diameter of the shaft. Both gears have to be modified, the small one should include a key to mount it on the shaft and the big one should have a bigger whole so the cable can pass through it.
Steel C45 might not be the most appropriate material for outdoors applications, so it is advisable to use stainless steel or other corrosion resistant material when the project comes out of the prototyping phase.
Gear wheels technical specifications:

- Material: Medium-carbon steel C45.
- Module: 3.
- Teeth: 70 and 12.
- Pitch diameter: 210 mm and 36 mm.
- Width: 30 mm.

2.6.7.6 Bearing

As it was pointed in section 2.6.1.1, the bearing Hiwin CRBE09025 is suitable for this project. The final bearing ordered was CRBE09025AWWC8P4, which includes a seal, threaded holes on inner ring, counter-bored holes on outer ring and the lowest accuracy grade available, since it is not considered critical.

Figure 2.26: Bearing Hiwin CRBE09025.

2.6.7.7 Motor controllers

The launch motor and the column have not changed, and the linear actuator has the same nominal current as the previous motor. Based on the same reasoning as in section 2.2.3.2, the motor controllers selected are RoboClaw 2x15A and RoboClaw 2x30A.

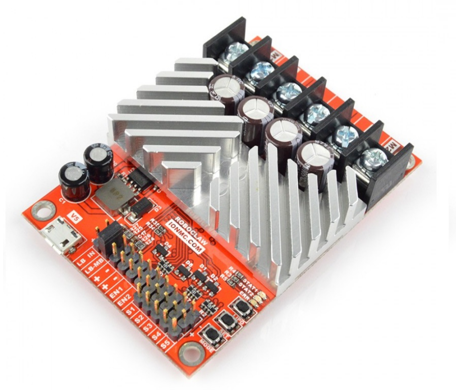


Figure 2.27: RoboClaw motor controller.

RoboClaw technical specifications:

- Current supported model 2x15A: 15 A per channel (30 A peak).
- Current supported model 2x30A: 30 A per channel (60 A peak).
- Voltage supported: 6 to 34 V.
- Current sensing.

2.6.7.8 Raspberry Pi

To control the launcher externally, establishing communication with the motor controllers, a Raspberry Pi 3 Model B+ is selected. It offers convenient possibilities thanks to its integrated Wifi antenna, and the motor controller also offer compatible libraries.

Figure 2.28: Raspberry Pi 3 Model B+.

2.6.7.9 Battery

The battery selected is a Multistar LiPo. The reasons to use this type of battery are their light weight and size (to mount it in the space left by the slider), high capacities and high discharge rates. Although the lifespan is shorter than in other batteries, it is acceptable for testing the prototype.

Figure 2.29: Battery Multistar 10000 mAh.

Multistar battery technical specifications:

- Capacity: 10000 mAh.
- Voltage: 22,2 V.
- Discharge rate: 10C.
- Peak discharge rate (10 seconds): 20C.
- Configuration: 6S.

Using the discharge rate and the capacity, the maximum current that the battery can provide can be calculated as:

$$Max\ battery\ current = C_{battery} \cdot capacity = 10 \cdot 10 = 100\ A \tag{2.45}$$

Assuming that all the motors of the prototype are working at the same time, it is necessary less than 100 A, which the battery can provide. However, this situation will never happen because the launch motor, the most demanding one in terms of current, will not work simultaneously with the others.

The capacity of 10000 mAh is also considered valid because it can power the linear actuator, the column and the rotation motor during 30 minutes, which is enough for testing purposes.

2.6.7.10 Cables and connectors

The battery selected comes with XT90 connectors, which are capable of supplying enough current. New cables soldered with these connectors will be used to connect all the components, taking advantage of the ease of soldering of this type of connector.

Regarding the cables, the ones that connect the battery with the launch motor are AWG 10 and the rest of them are AWG 14. Although the supported current for AWG 10 cables is slightly below 48 A, they can still be used because they should only keep this current for 0,2 seconds.

Figure 2.30: XT90 connector.

In order to connect the battery to more than one motor controller, the battery is plugged to a terminal block mounted on a DIN rail. By using this configuration, it is possible to add flexibility to the project, being possible to connect more modules in case that extra power connections are needed. For instance, in the future one more extra motor controller will be needed to control the opening mechanism of the capsule. Between the battery and the terminal block there is a switch to disconnect the system in case of emergency.

Figure 2.31: Terminal block Weidmüller WDU 10.

2.6.7.11 Metallic components

The rest of the metallic components have been mechanised by using a water cutter and a sheet of metal of 15 mm width. This was taken into account during the design so all the pieces could be obtained using this method.

3

Control

3.1 Introduction to the control method chosen

One of the main control objectives for this project is to be able to control the drone launcher using the closest device to the user. By doing that the time required to launch the drone is dramatically reduced, because for example, it would not be necessary to turn on a computer if the user is using an smartphone at that moment.

At this stage of the project, it is not considered important to use industrial equipment, and for that reason the components selected are very common and cheap, but enough for testing the prototype.

3.1.1 Electrical scheme

The prototype is powered by the battery, that supplies the energy to the to the components thanks to the terminal block where the Raspberry Pi and the RoboClaw motor controllers are plugged. Between the battery and the terminal block a switch is installed to turn on the prototype and to disconnect the energy supply in case of emergency.

Since all the components work with 24 V, it is not necessary to modify the voltage that the battery supplies. Only the Raspberry Pi needs the voltage to be reduced to 5 V, using a voltage regulator.

The first motor controller (RoboClaw 2x15A) is the master and it is connected directly to the USB port of the Raspberry Pi, while the second motor controller (RoboClaw 2x30A) is the slave and it is connected to the first one, using the ports S1 and S2. All communications are serial.

The first motor controller controls the linear actuator (port 1) and the rotation motor (port 2) and the second motor controller controls the lifting column (port 1) and the launch motor (port 2).

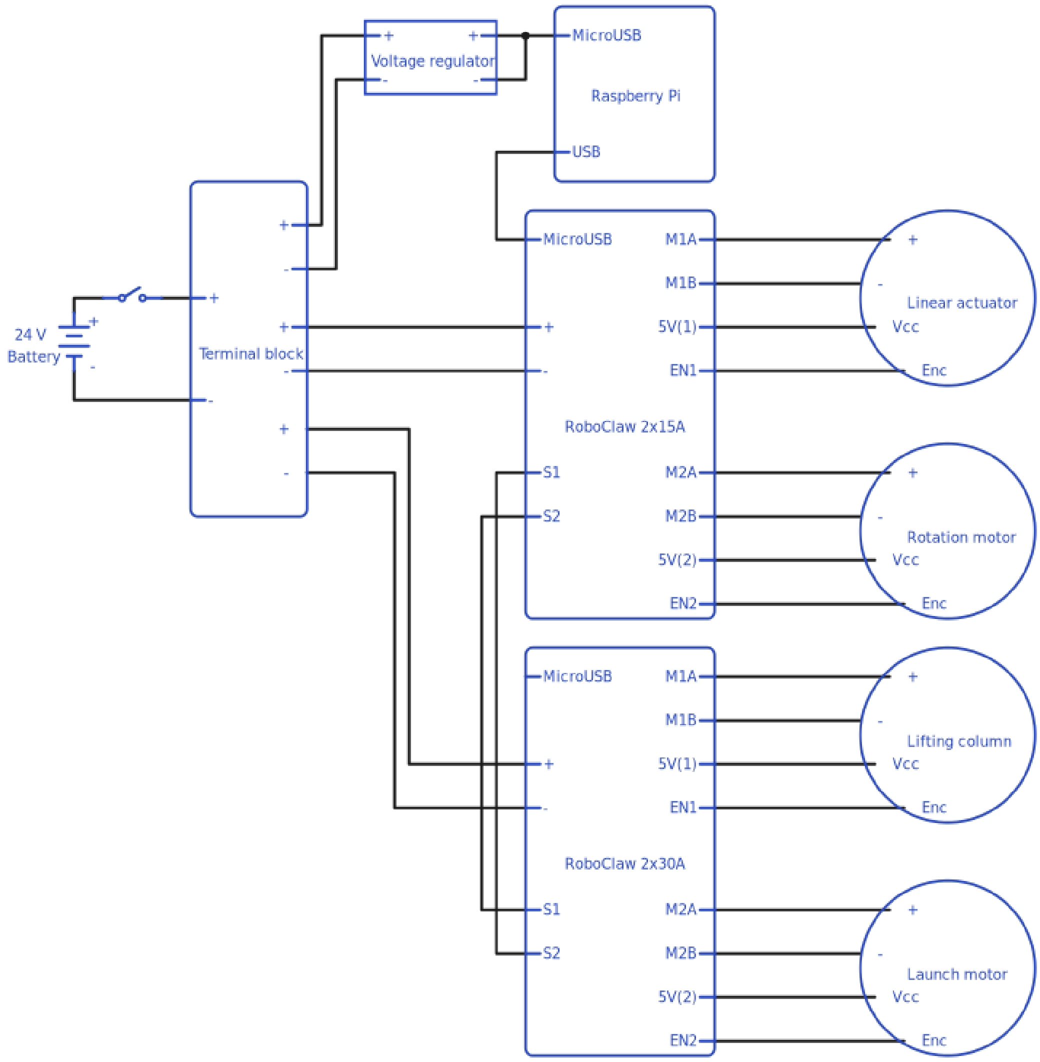


Figure 3.1: Electrical scheme.

3.1.2 Control scheme

First of all, to be able to establish a communication with the launcher, it is necessary that the launcher and the control device are in the same network. Using the built-in antenna of the Raspberry Pi it can be connected to the network without any addition peripherals.

Once the Raspberry Pi is connected, it deploys a server using its IP address and the port chosen. The server is created using a Python framework named Flask. This Python programme contains the necessary scripts to control the launcher and it also renders an HTML web page to all the devices accessing to the IP.

The web page, that now is accessible from different devices, contains several buttons, each one of them associated to a different Python script to perform the movements of the launcher. The web page also shows the live video from the camera installed in the launcher.

3.2 Configuration

First, it is necessary to configure the components to perform the desired control.

3.2.1 Raspberry Pi configuration

The Raspberry Pi runs Raspbian, a Debian-based computer operating system. Given that the libraries of the motor controller supplied are based on Python 2.7, this is the version used for the project.
The login credentials are:

- Username: pi.
- Password: dronelauncher.

3.2.1.1 Camera

The package UV4L should be installed using Linux repository. This package is used to stream the video obtained from the camera to the web page. The video is streamed into an IP address and a port and it can be included into the web page using the standard HTML commands.
The camera settings can be set with a web browser introducing the IP address of the Raspberry Pi followed by the port 8080.
It should configured with the following parameters:

- Resolution: 480x320.

3.2.2 RoboClaw motor controllers configuration

The RoboClaw configuration must be done through a software called *IonStudio* supplied by the manufacturer. Using this software, each motor controller should have the following parameters:

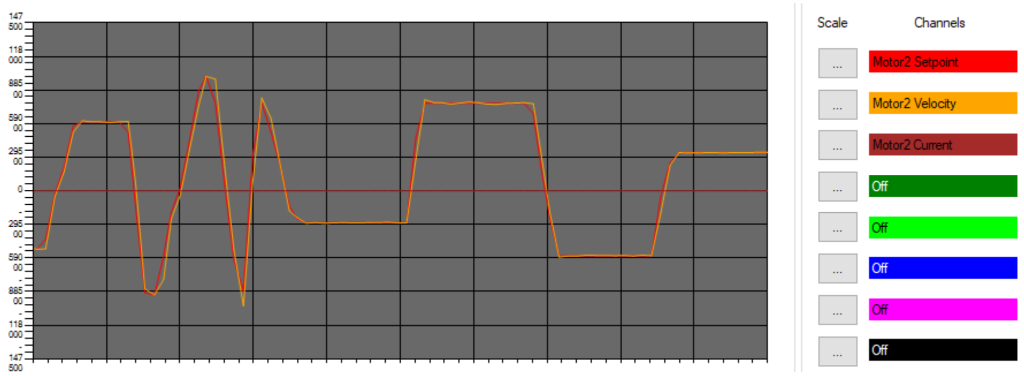
Table 3.1: Motor controller parameters

Motor controller 1 (RoboClaw 2x15A)	Control mode: Packet Serial. PWM mode: Sign Magnitude. USB-TTL Relay: Enabled. Packet serial address: 128. Baudrate: 115200. Battery cutoff: Auto Detect. M2 Reverse.
Motor controller 1 (RoboClaw 2x30A)	Control mode: Packet Serial. PWM mode: Sign Magnitude. USB-TTL Relay: Disabled. Packet serial address: 129. Baudrate: 115200. Battery cutoff: Auto Detect. M1 Reverse.

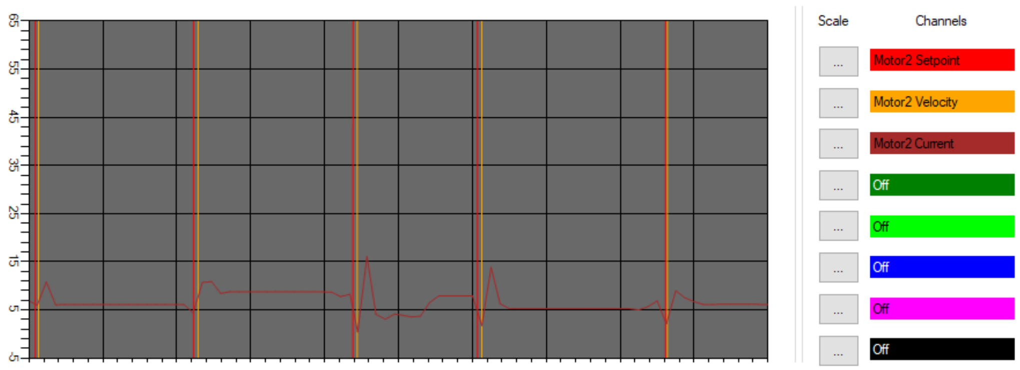
The rest of the settings can be used with their default values. As it can be seen, the only difference between them is the "USB-TTL Relay" and the "Packet serial address".
For the future, if a third motor controller is needed, it would be necessary to enable the "Multi-Unit Mode" on all the slaves and also add a pull-up resistor from 5 V to the line going to the master S1 pin.
Lastly, when the battery cutoff is set to "Auto Detect", the motor controller will stop working when the battery provides less than 18 V for the 6 cell battery that is being used (less than 3 V per cell). This is the absolute minimum voltage than a 6 cells LiPo battery can provide without being damaged, but it is recommended to charge it when the voltage reaches around 20,4 V (3,4 V per cell) to increase security and avoid malfunctioning.

3.2.3 PID tuning

To achieve a smooth transition between idle and the desired speed on every motor, it is necessary to tune a PID for each motor, which includes the typical PID parameters and QQPS (maximum speed of the motor in encoder pulses per second). These values should be introduced in the motor controllers using the *IonStudio* software to save them into the non-volatile memory. The values will be loaded after each power up so it is not necessary to define them every time the motor controllers are reset.



(a) Setpoint and speed during PID tuning



(b) Current during PID tuning

Figure 3.2: Launch motor PID tuning using *IonStudio*.

These PID are necessary to perform a closed loop speed control. The motor controller receive the amount of pulses per second and it increases or decreases the current to reach the objective. By doing that, it is possible to decide the desired speed for each movement and it will always move at that speed, no matter the battery voltage or the load of the launcher (new components could be added and it will behave like expected).

In order to tune the different PID, it has been tested that they are able to follow the speed setpoint when it is stable and also when it changes drastically. It is also important to check that the current does not oscillate too much, because that could damage the motors and be dangerous as well.

All the motors offered a secure and reliable performance, as can bee seen in figure 3.2, that corresponds to the launch motor. However, the behaviour of all the other motors is very similar to the one shown.

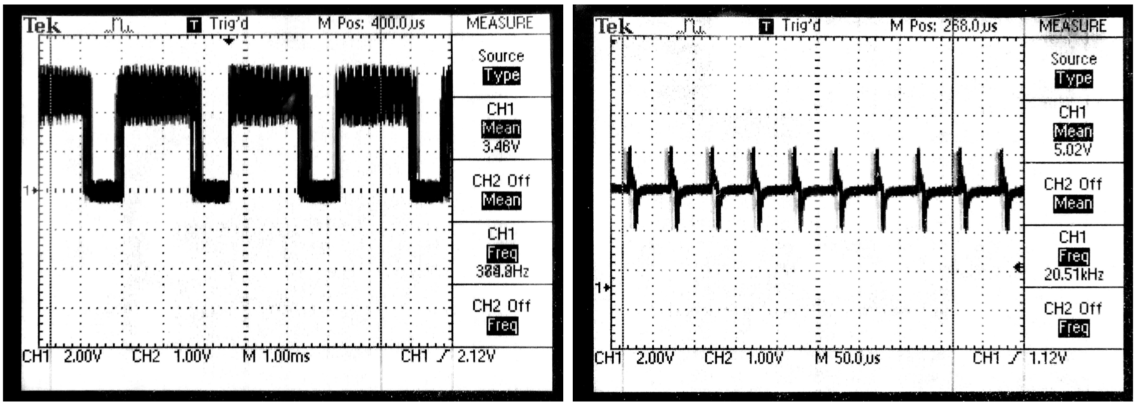
The PID parameters for the different motors are shown in the table below:

Table 3.2: PID parameters

Linear actuator	P: 1. I: 10. D: 0. QQPS: 7000.
Rotation motor	P: 1. I: 5. D: 0. QQPS: 240000.
Lifting column	P: 5. I: 10. D: 0. QQPS: 420.
Launch motor	P: 5. I: 10. D: 0. QQPS: 147500.

3.2.4 Encoder noise filter for the linear actuator

The motor controller was not able to count the pulses from the encoder of the linear actuator. After examining the signal using an oscilloscope, it showed high noise on both channels, making the control impossible. Taking a closer look it was determined that the noise had a frequency of 20 kHz. The noise and its frequency can be seen in figure 3.3.



(a) Noise (b) Noise frequency

Figure 3.3: Encoder noise and frequency of the noise using an oscilloscope.

For that reason, it was decided to design a simple RC low-pass filter with a cutoff frequency of 20 kHz whose diagram is shown in figure 3.4. It is important to remark that filtering the signal from an encoder can limit the maximum speed if the filter is too aggressive, and it can also make it unusable if the rising time is too long.

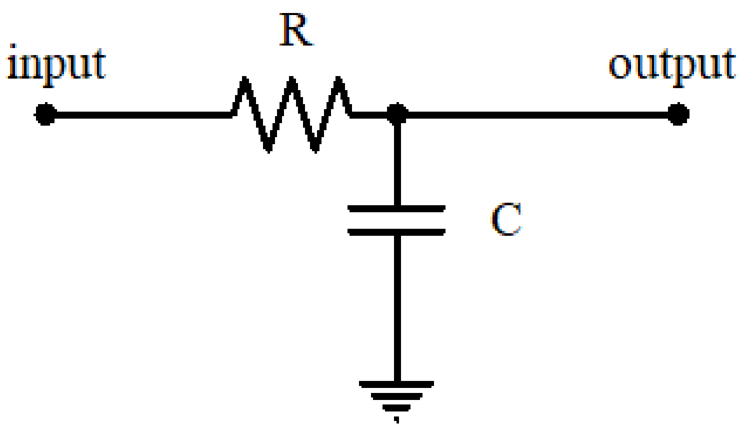


Figure 3.4: RC low-pass filter scheme.

Knowing the cutoff frequency and choosing a small capacitor (20 nF), the resistance can be calculated with the next equation:

$$f = \frac{1}{2\pi RC} \tag{3.1}$$

$$20000 = \frac{1}{2\pi R \cdot 1 \cdot 10^{-9}}$$

$$R = 8\ k\Omega$$

After selecting the components, the physical filter was mounted in a board, as it can be seen in figure 3.5.

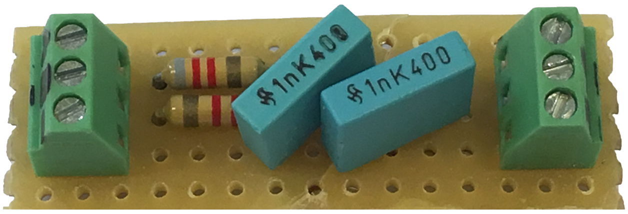


Figure 3.5: Filter.

As a result, the motor controller was able to read the filtered signal. After several tests, the results proved that the encoder did not have any accuracy problem and it was suitable to use it for controlling the speed and position of the linear actuator. The filtered signal is shown in figure 3.6

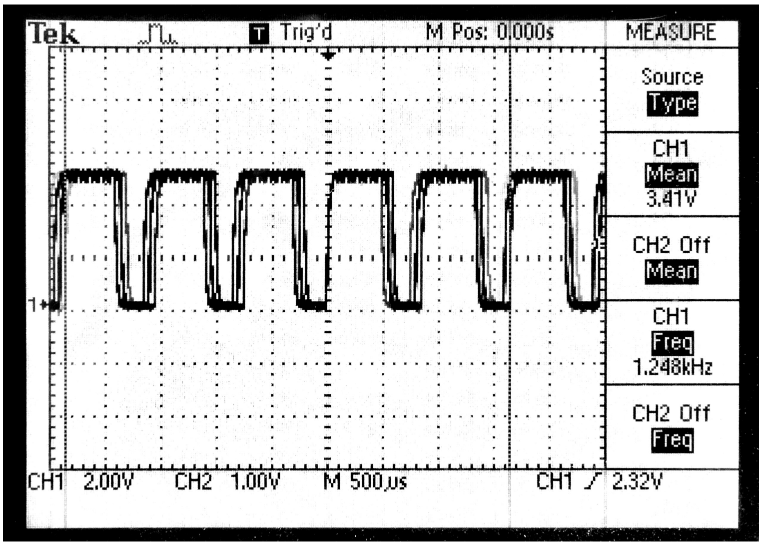


Figure 3.6: Noise filtered.

3.3 Interface

The interface has two different modes, manual and automatic. Besides, there is a video of the front part on the launcher to avoid launching when there are obstacles. The video is available in both modes.
The manual mode is used for testing purposes and it contains all the necessary buttons to control the launcher manually. Inside this mode there are different sections:

- Control motors individually: For every motor, it includes two buttons to move in two different directions and another one with a form to specify the absolute position where the launcher should move. Pitch and rotation should be specified in degrees and lift and launch in centimetres. This individual control does not have speed control, meaning that the motors will run at a given duty cycle instead of at a given encoder pulses per second. By doing this, this mode does not rely on the encoders and it can be used without them in case of malfunction.

- Max/min position: It has four buttons to move the launcher to the maximum and minimum positions of pitch and launch. There is not rotation or launch in this section because they are not considered necessary to have maximum and minimum positions, and their movement can be easily achieved with the other buttons.

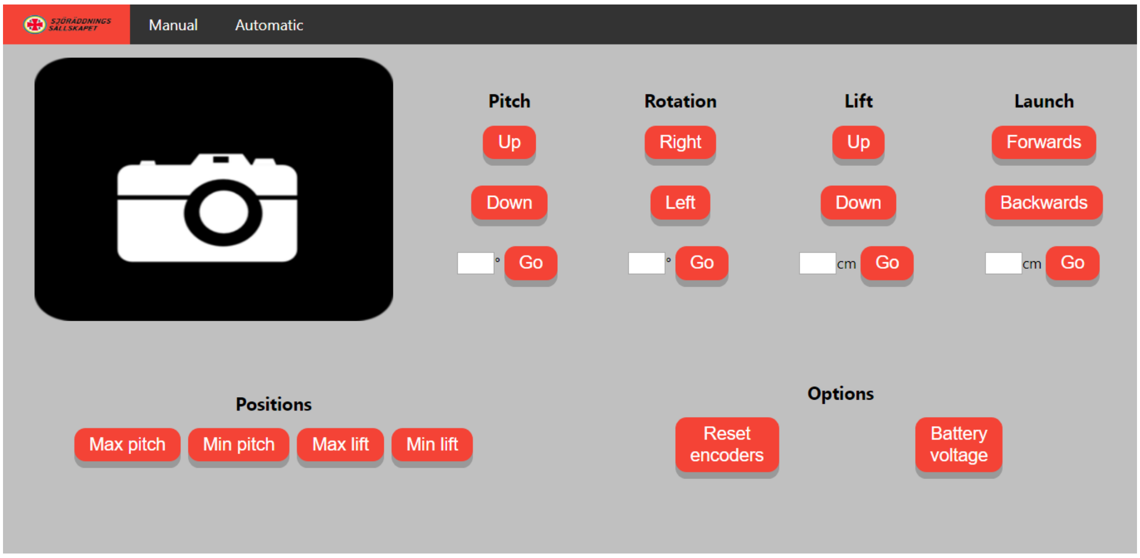
- Options: There are two buttons, reset the encoders (setting their home position where the motors are at that moment) and show battery voltage, which is useful to know when to charge the battery.

The automatic mode is intended to be the main mode for the final user. The objective is to keep it as simple as possible, with only the necessary buttons to perform a successful launch and mount the drone again. For that reason, apart from the video, there are only two sections:

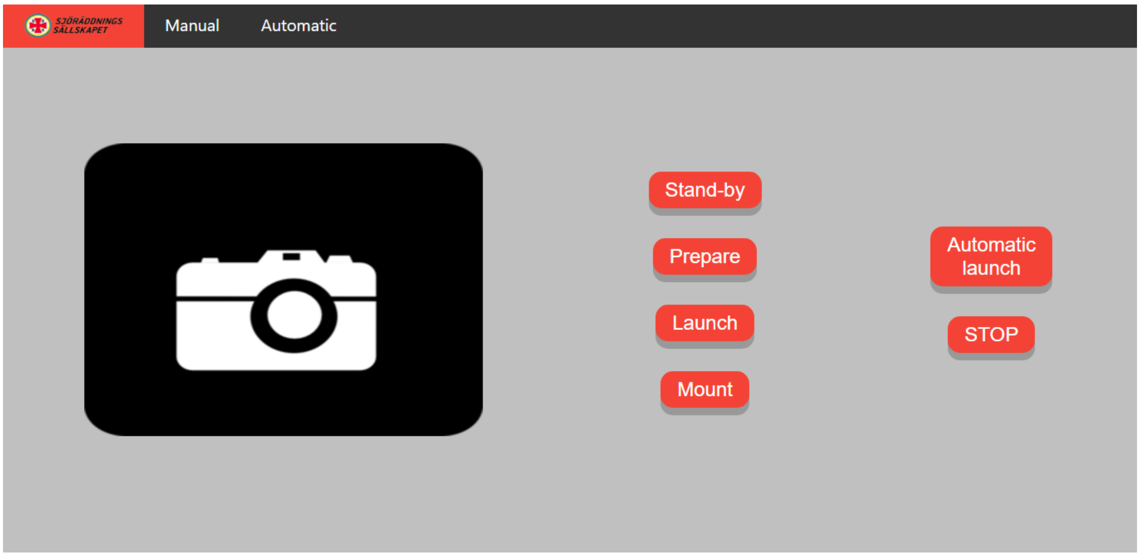
- Launch and rest positions: It includes four buttons, one for each position. The positions are stand-by (minimum pitch and lift, desired rotation and drone in the middle of the capsule), prepare (maximum lift, desired pitch and rotation and drone in the back part of the capsule), launch and mount (maximum pitch, minimum lift, desired rotation and platform ready to mount the drone).

- Automatic launch and stop: The automatic launch button should do everything it is required to launch the drone, which is, get ready, launch, and come back to the stand-by position. In this section there is also a Stop button to stop all the motors in case of emergency.

Figure 3.7 shows the interface running on a desktop web browser. The camera picture is used to display the location of the video.



(a) Manual mode



(b) Automatic mode

Figure 3.7: Manual and automatic mode of the control interface.

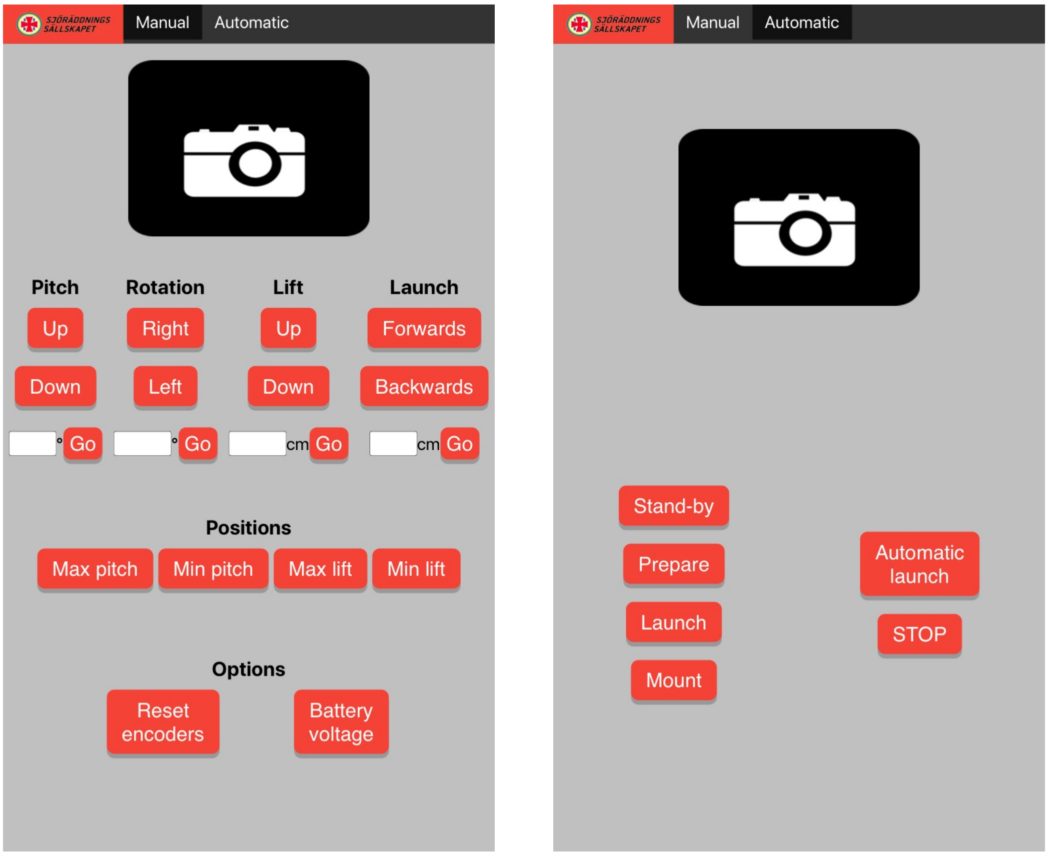
3.3.1 Code

The interaction between the user and the launcher is provided by a web page. It is all written in HTML but JavaScript is also necessary to offer a good experience. CSS is also used to format the elements and gather all the possible design options. All the code is commented to facilitate its understanding and its future expansion.

3.3.1.1 HTML and CSS

All the interface has been developed using common HTML commands. Since the web is going to be displayed in different devices, with different screen sizes, it should be as responsive as possible.

To achieve that, the interface has been divided in flex sections, that can change dynamically the horizontal and vertical space between them depending on the resolution available.
However, on vertical screens such as the portrait mode in smartphones, the HTML code developed does not show an adequate distribution. This is very important because usually the mobile devices are used in portrait mode, and forcing the users to use the landscape mode will lead to a poor interface design. To avoid that problem, Media Queries have been introduced in the CSS document. With this technique, it is possible to specify a different style for certain elements when the ratio of the screen is considered to be in portrait mode. The distribution of the elements on the screen is changed to use the space available in a better way, but all the control options are still available, as can be seen in figure 3.8.



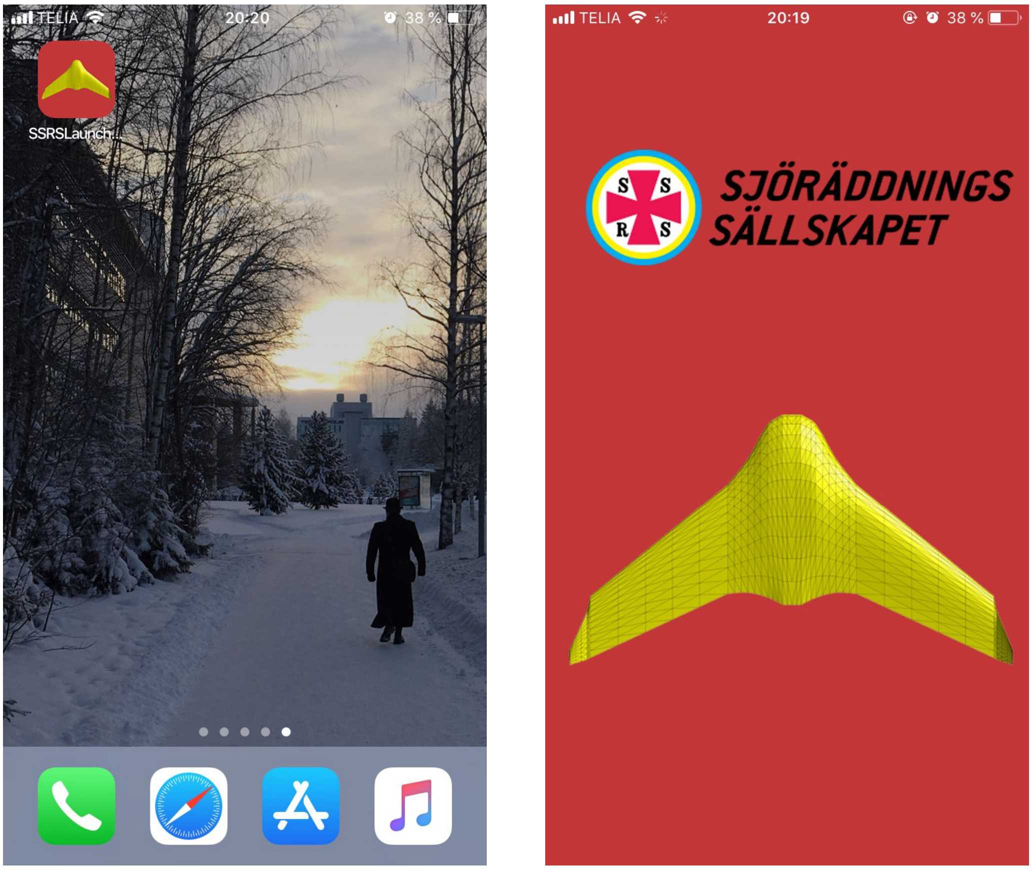
(a) Manual mode in portrait (b) Automatic mode in portrait

Figure 3.8: Manual and automatic mode of the control interface in portrait.

3.3.1.1.1 Progressive web application

Instead of having to introduce the IP address of the launcher on the web browser or having it stored as a bookmark, it is easier to access it through an application on the desktop, specially if it is being used on a mobile device.
By adding some HTML commands it is possible to turn the web page into a progressive web application, which allows it to be installed on the desktop with its own icon and a splash screen. Furthermore, if the web is accessed this way, it does not

display the browser interface, making it look like a native application on different operative systems like Android or iOS.



(a) Web application icon (b) Web application splash screen

Figure 3.9: Icon and splashscreen of the web application.

3.3.1.2 JavaScript

JavaScript is used to call the Python functions when the buttons are pressed, but a jQuery library is used to simplify the code. The buttons are prepared to be used with a computer and with a touch screen, using the different events mouse-up/mousedown and touchstart/touchend. Some web browsers, like Safari on iOS, triggered the mouse and the touch event, so it is necessary to include the command *event.preventDefault();* to prevent it.
But JavaScript is not only used for pressing the buttons, it is in charge of showing the elements depending on the selected mode (by default, manual mode is shown for testing purposes, but it can be changed). There is also a function to display the battery voltage and another one to hide or show the video on the interface, in case it is not necessary or the screen is too small to accommodate everything at the same time.
The usage of JavaScript is also necessary to offer a better user experience. For example, the manual controls of the motors need to be kept pressed to activate the movement, and the motor stops as soon as the button is no longer pressed. This

kind of user interaction is more natural than pressing a button to move and another one to stop.
Finally, it is remarkable the alert system, that prevents the user to introduce incorrect values in the forms, indicating with a pop-up which are the correct ones. This alert system also blocks the user from selecting a movement that uses the encoders, when the encoders are not reset and the launcher does not know in which position it is. To do that, every time the web is loaded it asks the launcher if the encoders are ready, and if they are not, it shows these buttons with a grey colour. Once the encoders are reset, the buttons return to their normal state.
All these checks, including the incorrect values and the encoders not being ready, are carried out on the server side, so the response is consistent on all the devices connected and it is harder to override by only using the browser, increasing the security. When the Python script finds an incorrect value, it returns an HTML status code that triggers the alert.

3.3.1.3 Python

While the rest of the code is executed in the user's browser, the Python code is executed on the server side, in this case, the Raspberry Pi. It is launched automatically after the Raspberry Pi is connected to a network.
As it has been exposed in section 3.1.2, the Python programme is used for two different things:

- Deploy a web server so the user can interact with the launcher using a web page.
- Send serial instructions to the motor controllers depending on the buttons pressed by the user on the web page.

The first one is achieved by using the framework Flask. After installing the package, it is only necessary to specify that the web page should be loaded by default and in which IP address and port the server should be set up. This IP address should match the address of the Raspberry Pi inside the local network, and the port can be chosen freely.
The rest of the Python code can be divided into:

- Import modules: Certain modules and libraries have to be imported, like Flask or the RoboClaw library.
- Open serial port: By doing that, the Raspberry Pi can send serial commands to the motor controllers.
- Declare variables: All the variables that should not change are declared here, like the encoder pulses and speed for each motor or the serial address for each motor controller.
- Functions: Each function uses the RoboClaw library to send serial packages to the motor controller depending on the movement that the user wants to do.

To exchange date between the server and the web page, like the voltage of the battery or the status of the encoders, JSON is used, so it is necessary to import the JSON module as well.
The Python serial instructions, obtained from RoboClaw library, and functions are summarised and explained in appendix B.

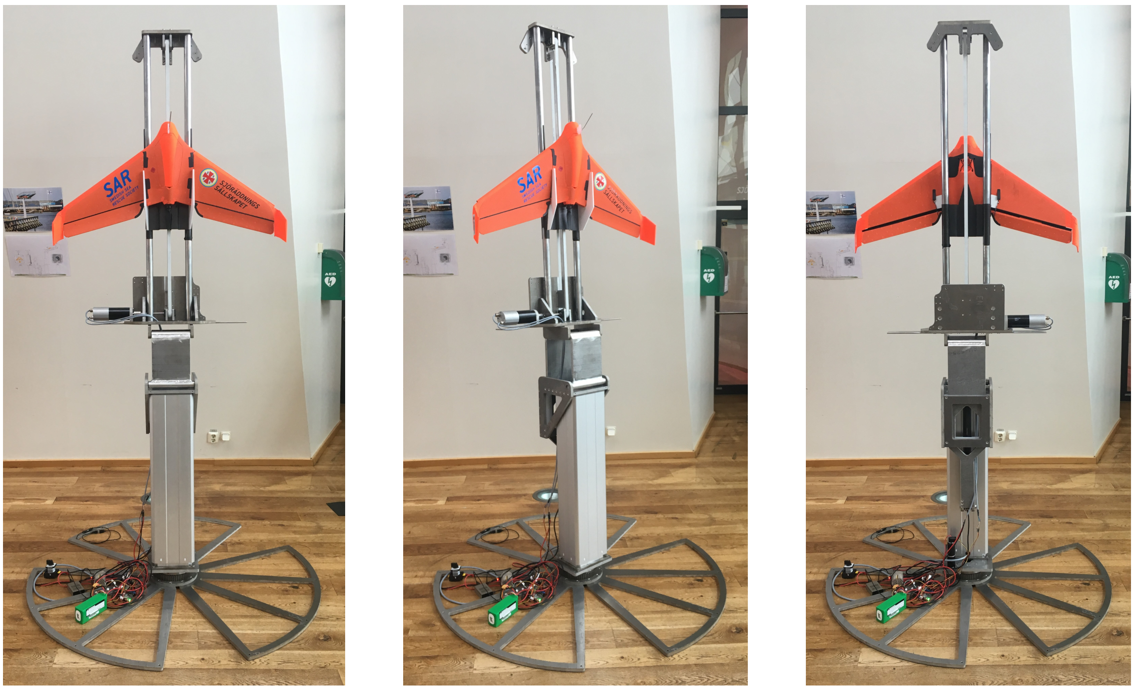
4
Results

4.1 Mounting the prototype

After receiving all the necessary components, they were mounted following the assembly. There were some mistakes that needed to be solved in collaboration with the mechanical workshop who mechanised the pieces.
The following sections will show the prototype in different positions to prove that it can reach them as it was expected.

4.1.1 Stand-by position

The stand-by position takes the column to its minimum height, the pitch to vertical position and the drone to the middle of the launcher to show it, as it has been specified before.

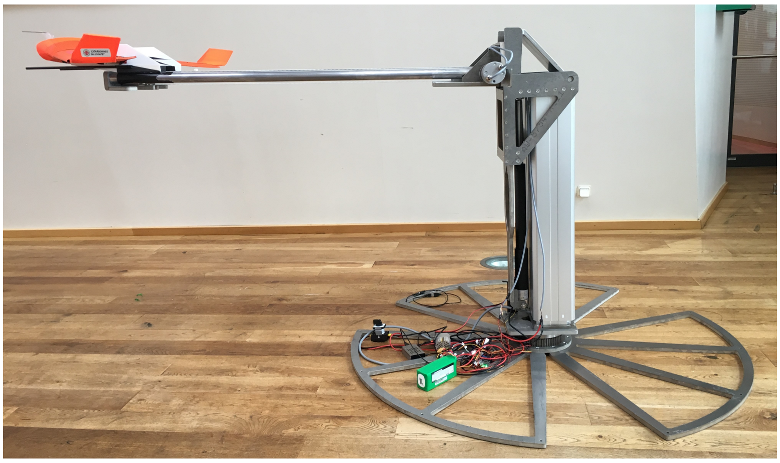


(a) Front view (b) Isometric view (c) Back view

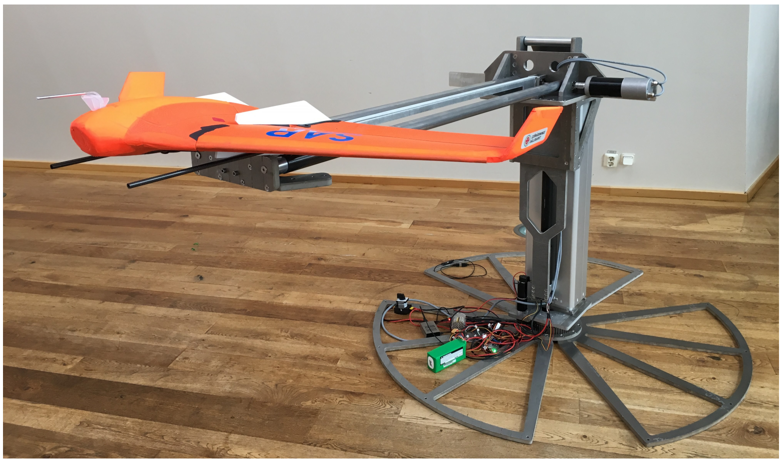
Figure 4.1: Prototype in stand-by position.

4.1.2 Mount position

The mount position, with the retracted column and maximum pitch, has been proved to be comfortable for an average person to return the drone to the platform.



(a) Front view



(b) Isometric view

Figure 4.2: Prototype in mount position.

4.1.3 Prepare position

For the prepare position, the column goes to its maximum length, the angle chosen is 45° and the rotation is 10°. These two last movements have been decided as an example but in the future they will need to be calculated depending on the wind speed and direction.

(a) Side view (b) Isometric view (c) Back view

Figure 4.3: Prototype in prepare position.

4.2 Differences from the drawings

Although most of the components are exactly like they were specified in the drawings, there are some small differences that are listed because they can be used in future iterations:

- Launch tube: It measures 141,1 cm instead of 140 cm because an old version of the drawings was used.
- Launch bottom back plate: Eight holes are for M8 instead of M6.
- Launch bottom plate: Because of the M8 holes in the previous component, 8,5 mm holes needed to be drilled to be able to mount both components. Also the holes for the belt bearings fixtures were drilled bigger (6,5 mm).
- Launch top plate: Same as before, the holes for the belt bearings fixtures needed to be drilled bigger (M6).
- Launch drive shaft: It was 1,2 cm too long so it was necessary to cut it. Also the key hole is just a flat surface as it can be seen in Figure 4.4 because the mechanical workshop did not have a tool small enough to do it. It is possible to mount the gear but it can fail at some point.

(a) Correct (b) Incorrect

Figure 4.4: Key hole.

5
Conclusion

5.1 Prototype improvements

After mounting the prototype, some problems have been found and should be corrected for a better final result:

- Impossible to reach 90°: Although it is very close, the linear actuator reaches its internal limit switch before arriving to 90°. One simple solution would be to mount the sliding plank a little higher using the bottom brackets fasteners.
- Three useless fasteners: Two fasteners from the component "Launch bottom plate" cannot be reached and they should be removed or moved to a different position. Also one of the fasteners in the "Launch motor fixture" does not have space and it should be moved.
- Belt access: One of the components that can be easily damaged is the belt, and to change it it is necessary to disassemble too many parts. It would be better to have an easier access to it.
- Keep drone in the stand-by position: Because of the weight of the drone, when the belt reaches the stand-by position, it goes down. It can be stopped using the motor, but it would mean maintaining the motor active all the time.
- Linear actuator bottom brackets: These components should have countersunk holes to reduce the space occupied by the bolt and use it to install a chain to organise the cables.
- Unify fasteners: Although it is not critical, it would be advisable to use the same type of fastener as long as it is possible. This would lead to a better design and repair experience, since it will not be necessary to have so many different fasteners in case something breaks.

5.2 Disadvantages of the control method

Although the control method chosen works well, it is possible to find different disadvantages that should be considered:

- Security: At this moment there is not any security measures implemented. To have access to the launcher it is only necessary to be inside the same network to which it is connected. This could be enough for testing the prototype but it should not be accepted in a definitive version.
- Connectivity: The launcher needs to be close to a Wi-Fi network in order to have access. In this case the prototype was always operating inside the range

of the SSRS connection, but it should not be assumed that these are going to be the circumstances in other locations.

- IP Address: Since it works with the IP address, it is necessary to know beforehand which is the address inside the network in order to deploy the web server. Based on that, the network should assign the same IP to the launcher to provide a reliable service, without changing parameters every time the router assigns a different IP.
- Web based: One of its strongest points is also one of its weaknesses. All the interface is based on web standards that worked at the moment of its development. However, it is possible that in the future new standards appear causing incompatibilities. Also it is optimised for current devices, but some changes on the future devices can lead to an unexpected behaviour.
- Industrial equipment: The Raspberry Pi and the RoboClaw motor controllers work perfectly for prototyping, but they are not made to be installed outdoors and they are not as reliable as other industrial equipment. Changing the components should be considered if the project reaches a more advanced phase.
- Interaction between interface and server: The interface based on web does not have any real time interaction with the server. This means that sometimes there could be some situations where the information on the screen does not reflect the actual state. For example, if two devices have opened the web, but the encoders are reset in one of them, the other device will not show that change unless the page is reloaded.

5.3 Future work

Apart from solving the existing problems, the project still need to add several functionalities before it is fully completed:

- Limit switches: In the actual state, the encoders need to be reset manually. By adding limit switches at the beginning and end of each movement, it would increase the safety, because the motors would stop when they reach them, and it would also be much easier to set the encoders to zero automatically. The motor controllers chosen have a mode that can be configured using their software and it provides automatic reset and home positioning every time the system is rebooted.
- Electronics box: All the electronic components should be organised in fixed positions, and also make the access to them as easy as possible. The best solution would be to design a box with defined spaces for each component to be installed next to the column, in the space reserved for that use. The cables should also be organised so they do not interfere when the column is retracted.
- Open/close capsule mechanism: The capsule and its opening and closing mechanism has not been designed yet. However, this part is fundamental for the launcher, because it protects the drone from any external damage, but also exposes it so the people can see it. The materials have to be chosen carefully in this case, because they need to be durable but also protect from the weather and vandalism.

- Design cover: The internal parts and electronics need to be covered to protect them and to provide a uniform and solid look to the launcher. It is important because the launcher is supposed to be installed outside, so it has to be safe for anyone approaching to it, solid to protect the internal components and good-looking to accomplish its marketing purposes.
- Turn away from the Sun: To avoid any kind of damage to the drone caused by the Sun, it would be interesting to develop a system where the launcher automatically turns away from the Sun depending on its position. Since the launcher is connected to internet, it would be possible to find an API that can provide that information and integrate it in the software.
- Weather data: To calculate the correct position to perform a successful launch, it is necessary to know the wind direction and speed. It would be possible to integrate some sensors in the launcher but it would be even more interesting to gather that information from the different sensors already installed in the area. To do that, the launcher should have access to the database of the sensors, but this is possible with the wireless connection available. Although the sensors are not installed in the exact position as the launcher, the necessary accuracy is not that high that it would be a problem.
- Solar energy: At the moment the prototype is powered by a battery, but it should have a more reliable system to be sure that the power is always available. One of the most interesting powering methods would be installing solar panels to recharge the batteries. By doing that, the launcher could be completely independent from the power grid and it could be installed in many different locations. Also the power consumption is not that high, so this solution seems technically feasible.

5.4 Implications and significance of the project

Even though the drone and its launcher are designed for sea rescues, the conclusions and the technology developed for them can be used for other purposes. The basic aim of the project is to have a drone ready to start under any condition, and that can be very helpful for things as variate as monitoring wild animals, helping the police to look for an objective or even as security control of large areas. Basically, the possible uses for a drone launcher like this are very wide, and they will be even more numerous if we attend to the increasingly interest of the society and the companies to incorporate drones to new tasks.
Apart from the possibility to adapt it to another environment, the knowledge acquired with the development of this prototype could also be used to improve its design in future iterations. This means that more details can be added to the launcher to improve its functionality, but using the basics developed on this thesis.

Bibliography

[1] Bloss, Richard (2014) *Unmanned Vehicles While Becoming Smaller and Smarter Are Addressing New Applications in Medical, Agriculture, in Addition to Military and Security.* Industrial Robot: An International Journal 41 (1): 82-86.

[2] S. P. Yeong, L. M. King, S. S. Dol (2015) *A Review on Marine Search and Rescue Operations Using Unmanned Aerial Vehicles.* International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering Vol. 9, No. 2.

[3] Farhan A. Salem (2013) *Mechatronics Design of a Solar Tracking System.* International Journal of Current Engineering and Technology Vol.3, No.3.

[4] Ola Delfin, Axel Nathorst-Westfelt, Lukas Riedel, Filip Svalander and Thure Waller (2015) *Utveckling av avfyrningshus för drönare.* Chalmers University of Technology.

[5] Grc.nasa.gov (2015) *Shape Effects on Drag.* [online] Available at: https://www.grc.nasa.gov/www/k-12/airplane/shaped.html [Accessed 23 Mar. 2018].

[6] Frank E. Jones (1978) *The Air Density Equation and the Transfer of the Mass Unit.* Journal of research of the Notional Bureau of Standards Vol. 83, No. 5.

[7] Maxemow, Shane (2009) *That's a Drag: The Effects of Drag Forces.* Undergraduate Journal of Mathematical Modeling One + Two Vol. 2, Iss. 1, Article 4.

[8] Weatherspark.com (2018) *Average Weather in Göteborg.* [online] Available at: https://weatherspark.com/d/71566/1/5/Average-Weather-on-January-5-in-G% C3% B6teborg-Sweden# Sections-Wind [Accessed 23 Mar. 2018].

[9] F. Živića, M. Babića, S. Mitrovića, D. Adamovića, S. Pelemisb (2014) *Friction Coefficient of UHMWPE During DryReciprocating Sliding.* Tribology in Industry Vol. 36, No. 3 281-286.

[10] Shoufan Cao, Hongtao Liu, Shirong Ge and Gaofeng Wu (2011) *Mechanical and tribological behaviors of UHMWPE composites filled with basalt fibers.* Journal of Reinforced Plastics and Composites 30(4) 347–355.

[11] Fernando Lasagni and Hans Peter Degischer (2010) *Enhanced Young's Modulus of AlSi Alloys and Reinforced Matrices by Co-continuous Structures.* Journal of COMPOSITE MATERIALS, Vol. 44, No. 6.

[12] Beléndez Tarsicio, Neipp Cristian, Beléndez Augusto (2002*) Large and small deflection of a cantilever beam.* European Journal of Physics Vol. 23, No. 3.

[13] Goutam Saha and Sajeda Banu (2007) *Buckling load of a beam-column for different end conditions using multi-segment integration technique.* ARPN Journal of Engineering and Applied Sciences Vol. 2, No. 1.

[14] Hiwin (2014) *Crossed Roller Bearings Technical Information.* [online] Available at: https://www.hiwin.com/pdf/crossed_roller_bearings.pdf [Accessed 11 Apr. 2018].

A

Budgets

This appendix includes the two different budgets, the first version of the launcher, with the worm gearbox, and the final version that ended up being built. A comparison between them is also included.

A.1 Worm gearbox mechanism budget

All the necessary components that can be bought are listed below, grouping them by supplier. The components that need to be mechanised are not included.

Table A.1: Worm gearbox launcher budget

Component	**Name**	**Items**	**Price/item (SEK)**	**Supplier**
Lifting column and encoder	DB	1	25235	X2 Technology
Belt wheel	27AT5/60-0	1	622,5	Mekanex
Belt wheel	27AT5/12-2	1	247,5	Mekanex
Belt	16-T5-460	1	263	Mekanex
Axis-clamps	172-427547	2	333,75	Mekanex
Worm gearbox	B 110 FB 53,00 C D -M V5	1	10000	Mekanex
	Shipping		262,5	Mekanex
Rotation motor	RE 40 Ø40 mm, Graphite Brushes, 150 Watt (Part No.: 148867)	1	4121,25	Maxon
Gearbox	Planetary Gearhead GP 42 C Ø42 mm, 3 - 15 Nm, Ceramic Version (Part No.: 203124)	1	2946,25	Maxon
Encoder	Encoder HEDS 5540, 500 CPT, 3 Channels (Part No.: 110513)	1	861,25	Maxon
Pitch motor	RE 40 Ø40 mm, Graphite Brushes, 150 Watt (Part No.: 148867)	1	4121,25	Maxon

Gearbox	Planetary Gearhead GP 42 C Ø42 mm, 3 - 15 Nm, Ceramic Version (Part No.: 203122)	1	2946,25	Maxon
Encoder	Encoder HEDS 5540, 500 CPT, 3 Channels (Part No.: 110513)	1	861,25	Maxon
	Shipping		327,5	Maxon
Bearing	THK RU85	1	1420	Aliexpress
Wind sensor	Direction (Voltage type)	1	415	Aliexpress
Arduino	Uno	1	212	RobotShop
Motor controller 1	RoboClaw 2x30A	1	1250	RobotShop
Motor controller 2	RoboClaw 2x15A	1	900	RobotShop
	Shipping		253	RobotShop
Total			**57933**	

A.2 Final version budget

This new budget includes more details that the one in table A.1, introducing also the necessary electrical components to connect and power the launcher. Nevertheless, some minor components such as USB cables or spade connectors have not been included due to its lack of importance in the final price. Changes in suppliers are due to stock availability differences in the moment of the order. Also, the components from the mechanical workshop are not included.

Table A.2: Vertical linear actuator launcher budget

Component	**Name**	**Items**	**Price/item (SEK)**	**Supplier**
Lifting column and encoder	DB	1	25235	X2 Technology
Big gear wheel	M3X70-P	1	1391	OEM Motor
Small gear wheel	M3x12	1	81	OEM Motor
Bearing	Hiwin CRBE09025AWWC8P4	1	5602	Aratron
	Shipping		1000	Aratron
Rotation motor	RE 40 Ø40 mm, Graphite Brushes, 150 Watt (Part No.: 148867)	1	4121,25	Maxon
Gearbox	Planetary Gearhead GP 42 C Ø42 mm, 3 - 15 Nm, Ceramic Version (Part No.: 203124)	1	2946,25	Maxon
Encoder	Encoder HEDS 5540, 500 CPT, 3 Channels (Part No.: 110513)	1	861,25	Maxon

Launch motor	RE 65 Ø65 mm, Graphite Brushes, 250 Watt (Part No.: 388984)	1	7425	Maxon
Encoder	Encoder HEDL 9140, 500 CPT, 3 Channels (Part No.: 386001)	1	1862,5	Maxon
Linear actuator	ANT-52, 250 mm stroke, Hall sensor	1	2000	Sito-Motor
Motor controller 1	RoboClaw 2x30A	1	1104	Pololu
Motor controller 2	RoboClaw 2x15A	1	795	Pololu
	Shipping		604	Pololu
Battery	Multistar 10000 mAh 6S 10C	1	619,75	HobbyKing
Cable	6 m 10 AWG Black	1	128	HobbyKing
Cable	5 m 10 AWG Red	1	132	HobbyKing
Cable	10 m 14 AWG Black/Red	1	106	HobbyKing
Connectors	XT90 male/female	1	50,3	HobbyKing
Raspberry Pi	Model 3 B+	1	416,25	Elfa
SD card	MicroSD 16 GB	1	112	Elfa
Camera	HD-Wide-Angle Camera Module	1	413,75	Elfa
Terminal block	Weidmüller WDU 10	6	40,75	Elfa
Terminal block	Cross connector	2	22,87	Elfa
Terminal block	End bracket	2	19,12	Elfa
Terminal block	End plate	1	11,97	Elfa
Terminal block	DIN rail	1	81,12	Elfa
Switch	Main switch 13 kW	1	311,25	Elfa
Total			**57739,12**	

A.3 Comparison

Comparing the price of the previous pitch mechanical components (18256,25 SEK) with the new one, where only a linear actuator is needed (2000 SEK), the price has been reduced by around 90 %. Regarding the rotation, the motor and gearbox is the same but the price of the bearing has increased because it is bigger and the quality is higher to support the momentum from the base. Even including more components from the electrical part and the new launch motor, the total price is now lower than in the previous budget.

B
Python

This appendix gathers all the serial instructions sent by the Raspberry Pi to the motor controller and the functions in Python. The serial instructions use the RoboClaw library supplied by the manufacturer.

B.1 Serial instructions

`rc = Roboclaw("/dev/ttyACM0",115200)`
`rc.Open()`
Opens the serial port with a speed of 115200 bauds.

`rc.ForwardM1(address, speed)` / `rc.ForwardM2(address, speed)`
Drives motor 1 / motor 2 forward, specifying the address of the motor controller and with an speed range from 0 to 127. It is also used for stopping the motor choosing 0 speed.

`rc.BackwardM1(address, speed)` / `rc.BackwardM2(address, speed)`
Drives motor 1 / motor 2 backward, specifying the address of the motor controller and with an speed range from 0 to 127.

`rc.SpeedDistanceM1(address,speed,distance,buffer)` /
`rc.SpeedDistanceM2(address,speed,distance,buffer)`
Drives motor 1 / motor 2 with a signed speed (encoder pulses per second) for the given distance (encoder pulses). The sign on the speed and distance indicates which direction the motor will run.
The Buffer can be set to a 1 or 0. If a value of 0 is used the command will be buffered and executed in the order sent. If a value of 1 is used, the current running command is stopped, any other commands in the buffer are deleted and the new command is executed. Each motor has separate buffers.

`rc.ReadEncM1(address)[1]` / `rc.ReadEncM2(address)[1]`
Reads encoder for motor 1 / motor 2. The parameter "[1]" reads only the actual encoder pulse (position).

`rc.ResetEncoders(address)`
Resets encoders to zero.

`rc.ReadMainBatteryVoltage(address)[1]`
Reads the battery voltage in 10ths of a volt.

B.2 Functions

`index()`
Renders the HTML template.

`function_movement_direction()`
Movement can be pitch, rotation, lift or launch and direction can be up or down.
Drives the motor selected forward or backward.

`function_movement_position()`
Movement can be pitch, rotation, lift or launch.
Checks that the encoders have been reset and receives the desired position. Then, it calculates the encoder pulses that the motor should move based on the total encoder pulses and the desired position. Finally, moves the motor forward or backward to reach the objective.

`function_movement_stop()`
Movement can be pitch, rotation, lift or launch.
Stops the motor.

`function_max_movement()` / `function_min_movement()`
Movement can be pitch or lift.
Drives the motor to the maximum or minimum position.

`function_reset_encoders()`
Resets encoders to zero.

`function_battery_voltage()`
Gets the battery voltage, turn the value into volts and rounds it to 2 decimals. The result is returned using JSON.

`function_stop()`
Stops all the motors.

`function_standby()`
Checks that the encoders have been reset. Then, it calculates the encoder pulses that the motors should move based on the total encoder pulses and the actual position. Finally, moves the motors forward of backward to reach the objectives.
The objectives are minimum pitch, minimum lift, desired rotation and drone in the middle of the capsule.

`function_prepare()`
Checks that the encoders have been reset. Then, it calculates the encoder pulses that the motors should move based on the total encoder pulses and the actual position. Finally, moves the motors forward of backward to reach the objectives.
The objectives are maximum lift, desired pitch and rotation and drone in the back part of the capsule.

`function_launch()`
Checks that the encoder from the launch motor have been reset. Then, it calculates the encoder pulses that the motor should move to go to the back part of the capsule, based on the total encoder pulses and the actual position. Finally, drives the launch motor to launch the drone.

`function_mount()`
Checks that the encoders have been reset. Then, it calculates the encoder pulses that the motors should move based on the total encoder pulses and the actual position. Finally, moves the motors forward of backward to reach the objectives.
The objectives are maximum pitch, minimum lift, desired rotation and platform ready to mount the drone.

`function_automatic_launch()`
Checks that the encoders have been reset. Then, it calculates the encoder pulses that the motors should move based on the total encoder pulses and the actual position. Finally, moves the motors forward of backward to reach the objectives.
The objectives are first, go the launch position, then, launch the drone, and finally, go to stand-by position.

`function_disable_buttons()`
Returns the variable encoders_ready, which indicates if the encoders have been reset, using JSON.

C

Drawings

This appendix collects all the drawings used in the project to mechanise the metallic and plastic components necessary to build the prototype. These blueprints were drawn by Fredrik Falkman.

The reason to include them is to make this document an accurate compilation of all the work performed related with the drone launcher and also to indicate all the real measures, which can be useful in the next iterations of the project.

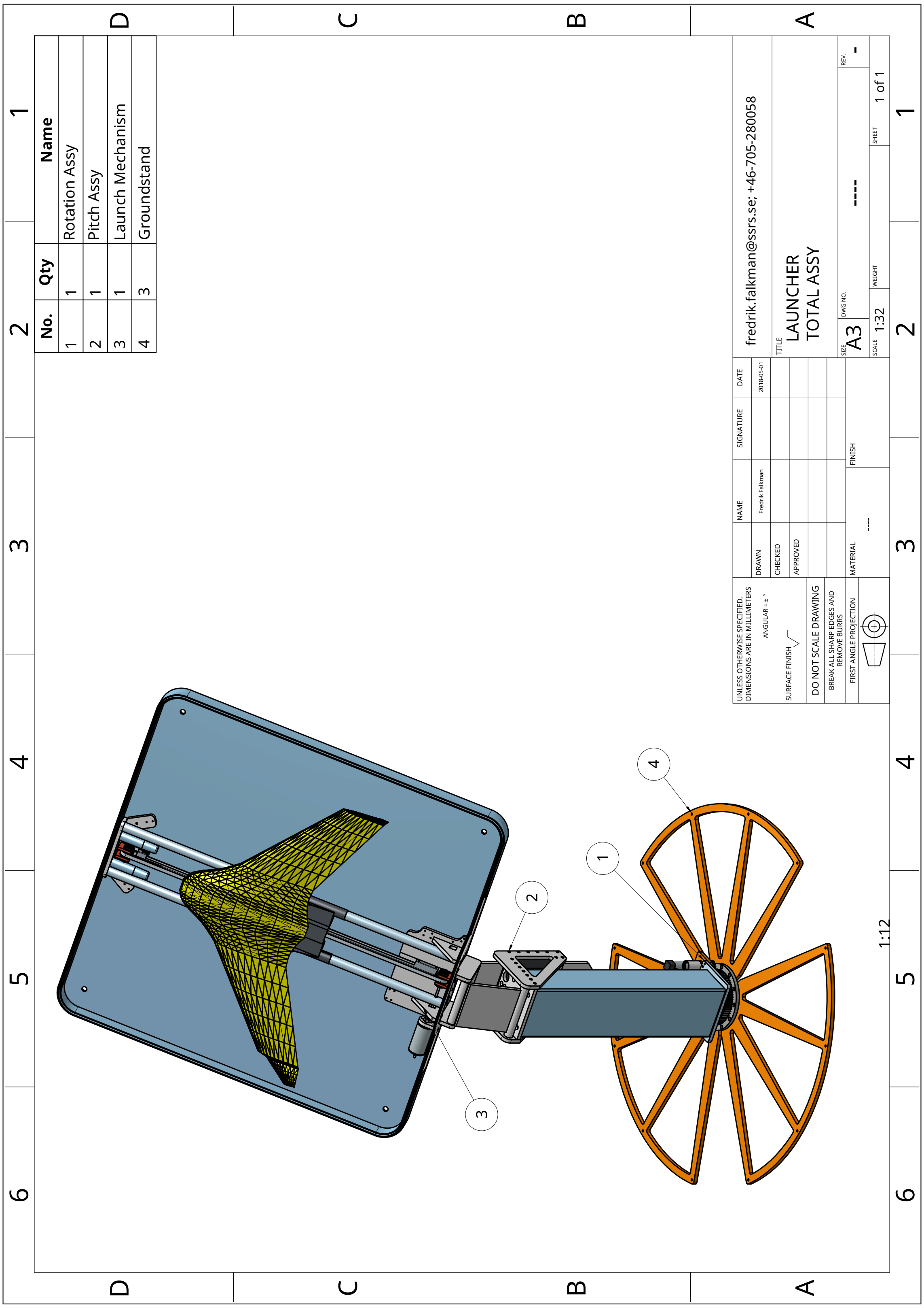
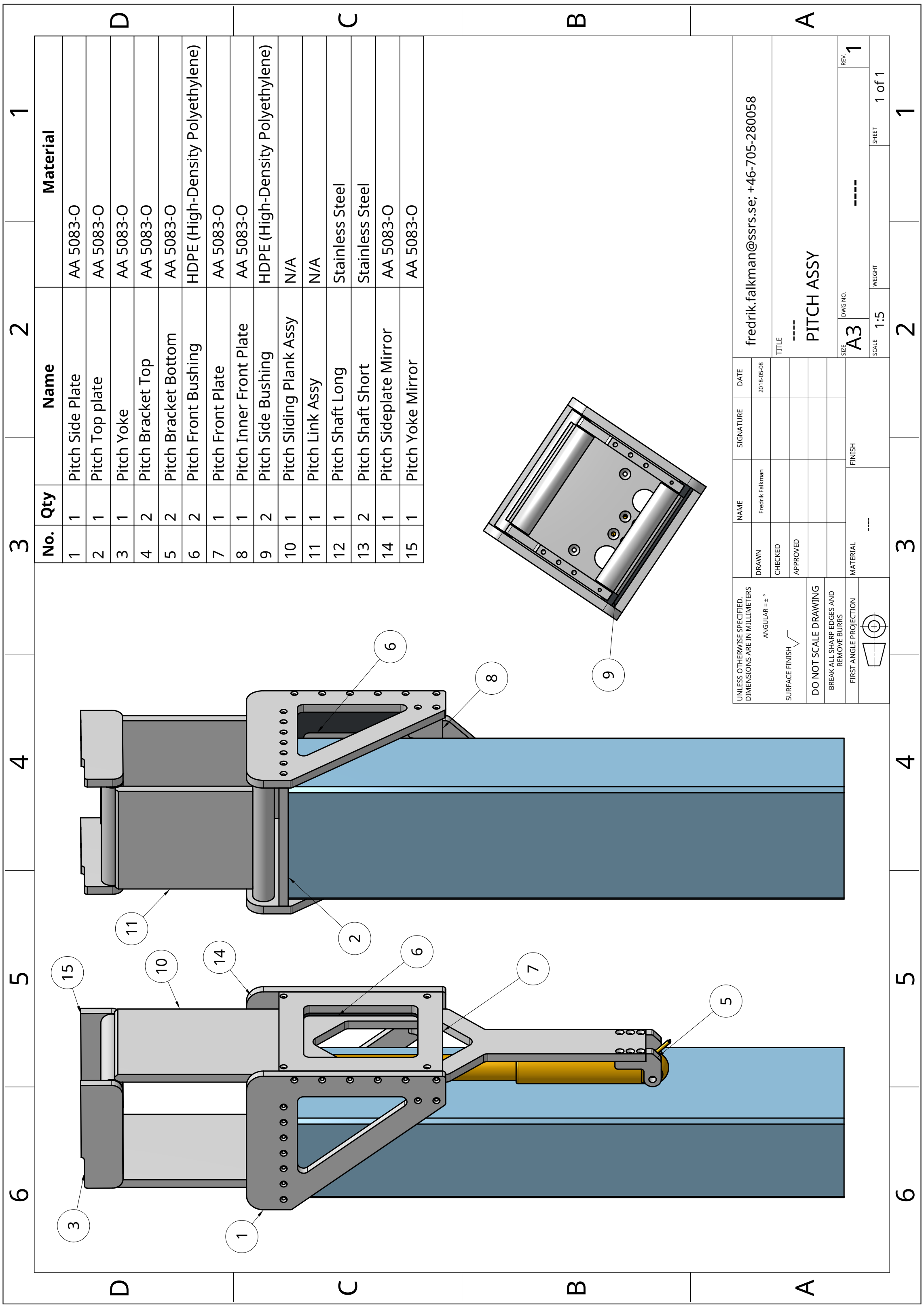
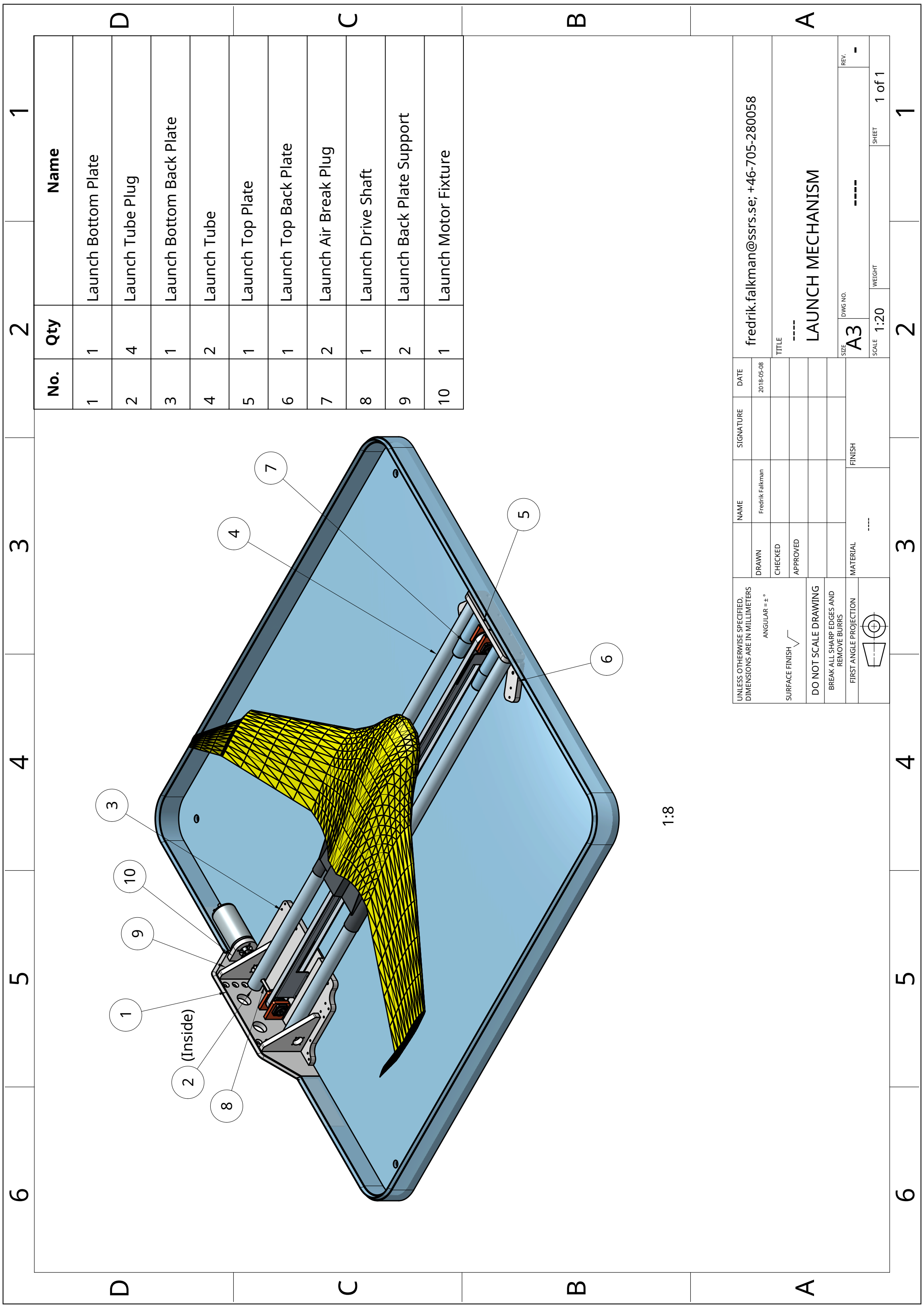


Figure C.1: Total assembly.



No.	Qty	Name	Material
1	1	Pitch Side Plate	AA 5083-O
2	1	Pitch Top plate	AA 5083-O
3	1	Pitch Yoke	AA 5083-O
4	2	Pitch Bracket Top	AA 5083-O
5	2	Pitch Bracket Bottom	AA 5083-O
6	2	Pitch Front Bushing	HDPE (High-Density Polyethylene)
7	1	Pitch Front Plate	AA 5083-O
8	1	Pitch Inner Front Plate	AA 5083-O
9	2	Pitch Side Bushing	HDPE (High-Density Polyethylene)
10	1	Pitch Sliding Plank Assy	N/A
11	1	Pitch Link Assy	N/A
12	1	Pitch Shaft Long	Stainless Steel
13	2	Pitch Shaft Short	Stainless Steel
14	1	Pitch Sideplate Mirror	AA 5083-O
15	1	Pitch Yoke Mirror	AA 5083-O

Figure C.2: Pitch assembly.



No.	Qty	Name
1	1	Launch Bottom Plate
2	4	Launch Tube Plug
3	1	Launch Bottom Back Plate
4	2	Launch Tube
5	1	Launch Top Plate
6	1	Launch Top Back Plate
7	2	Launch Air Break Plug
8	1	Launch Drive Shaft
9	2	Launch Back Plate Support
10	1	Launch Motor Fixture

Figure C.3: Launch assembly.

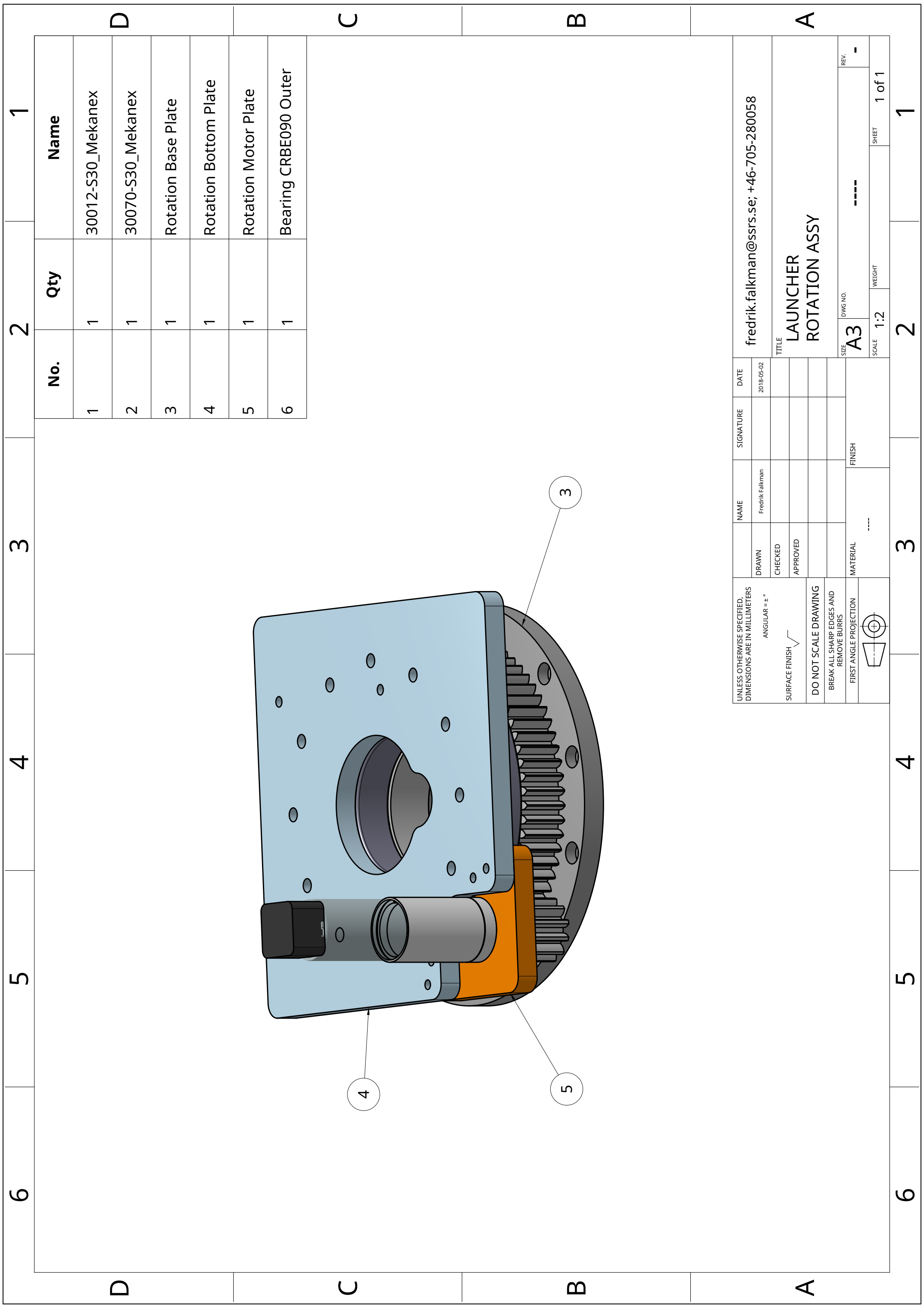


Figure C.4: Rotation assembly.

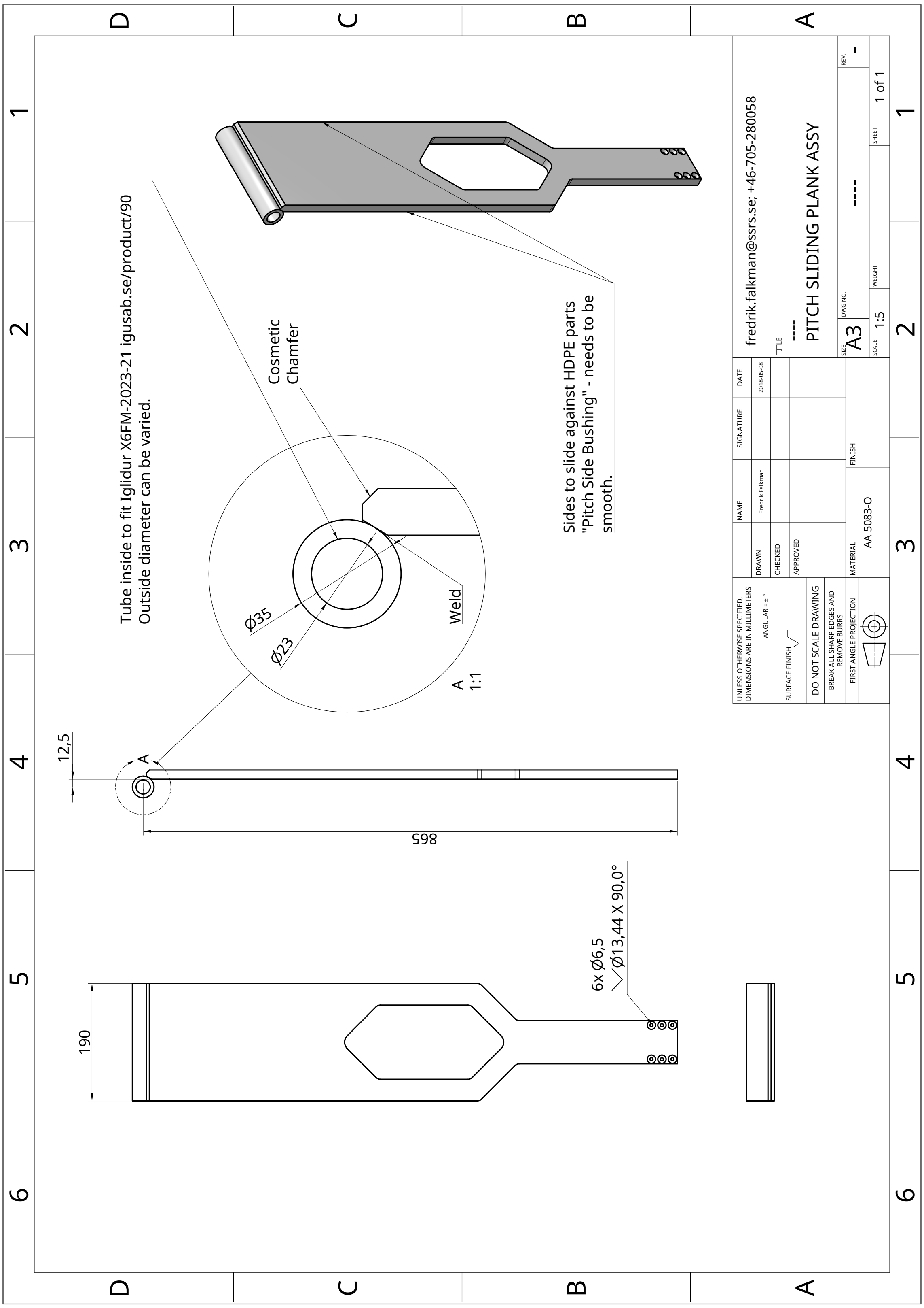


Figure C.5: Pitch sliding plank.

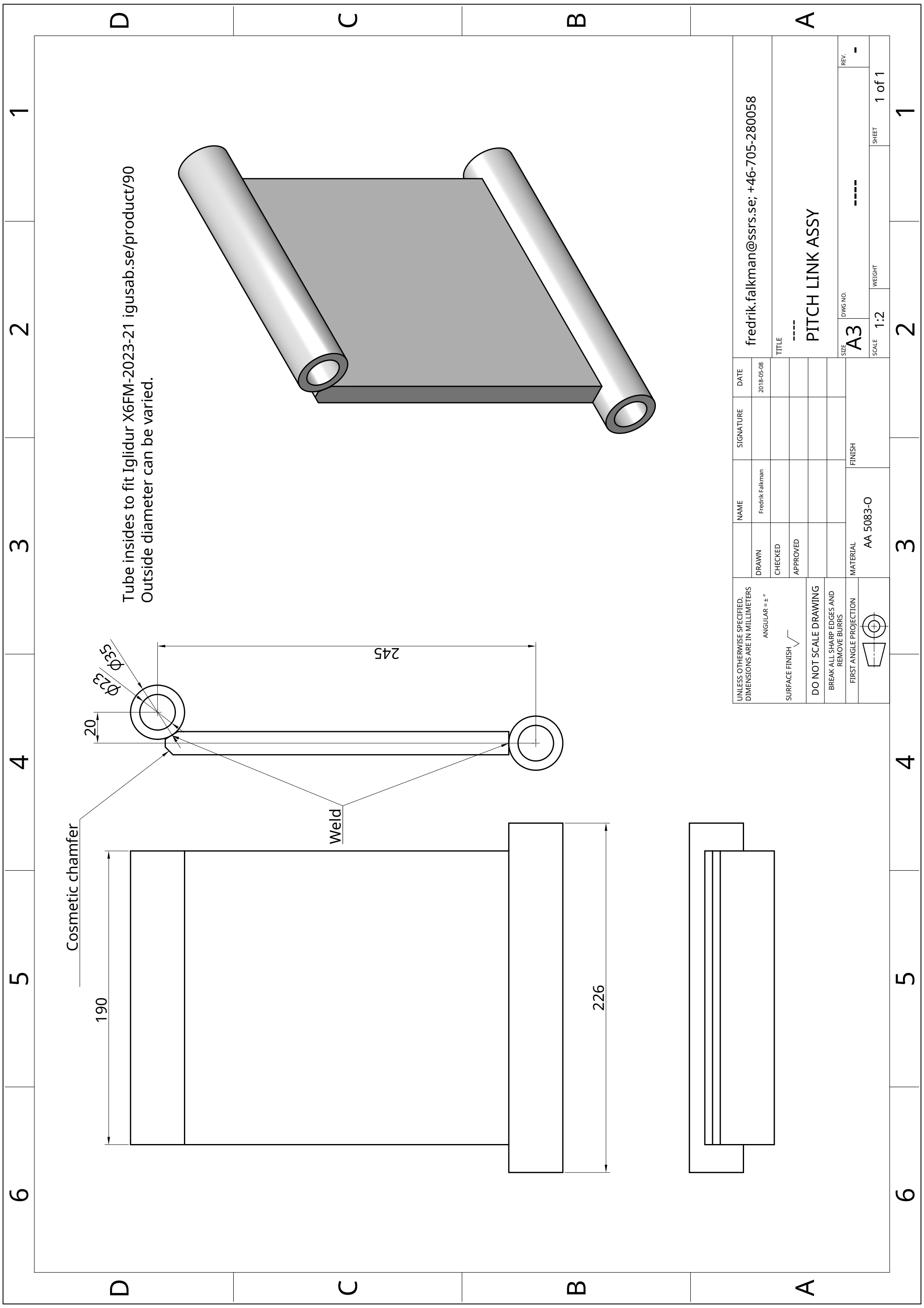


Figure C.6: Pitch link.

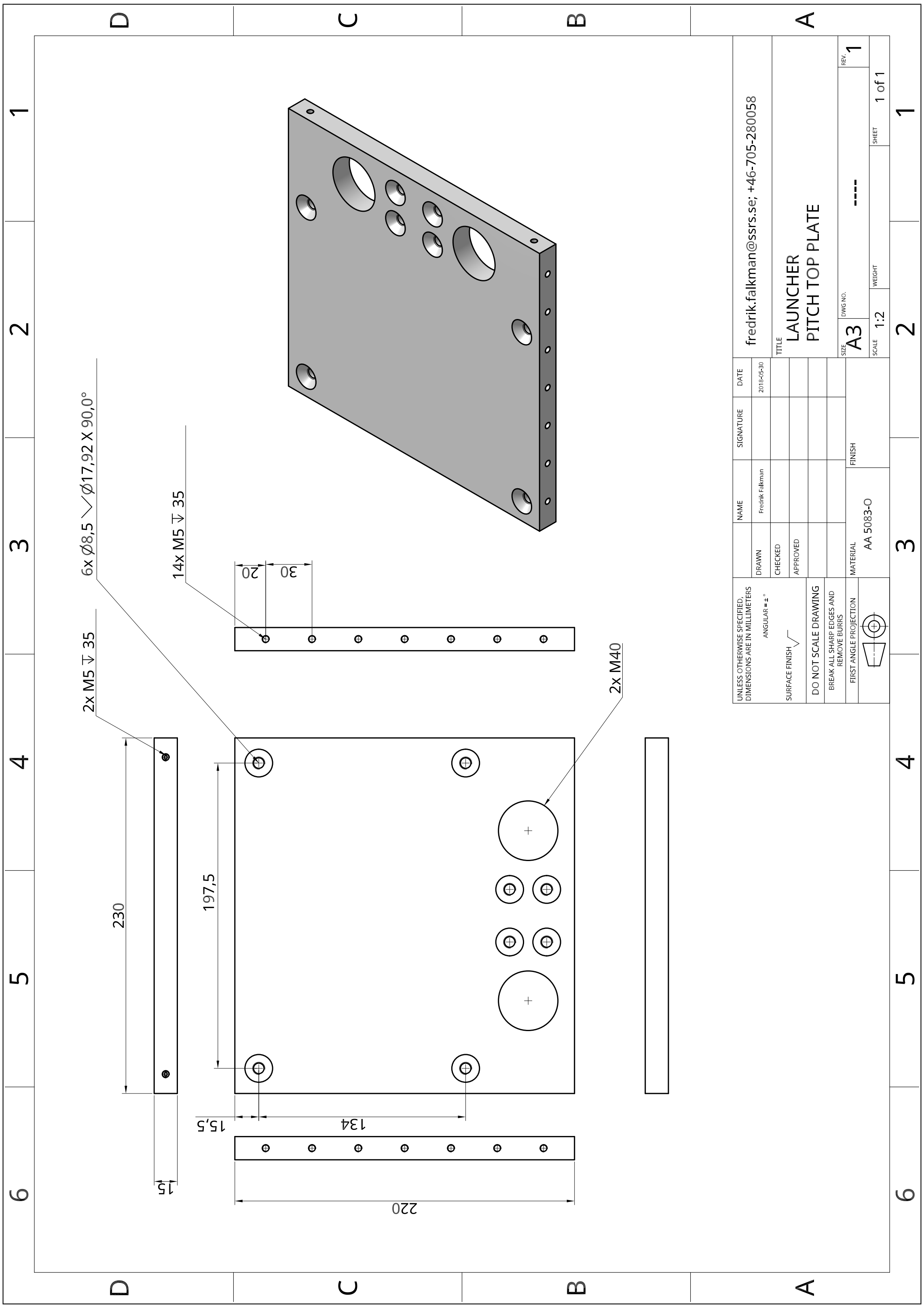


Figure C.7: Pitch top plate.

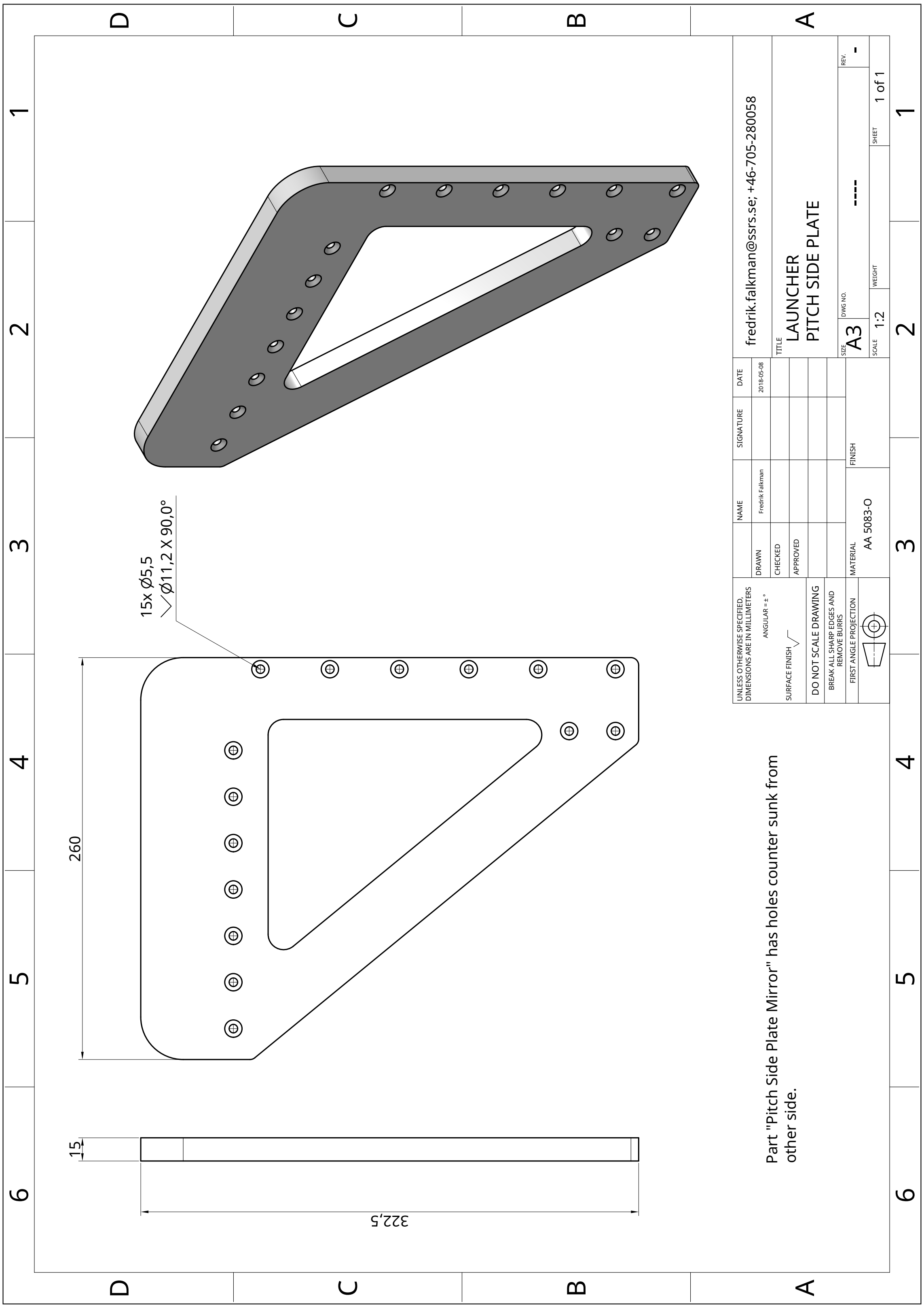


Figure C.8: Pitch side plate.

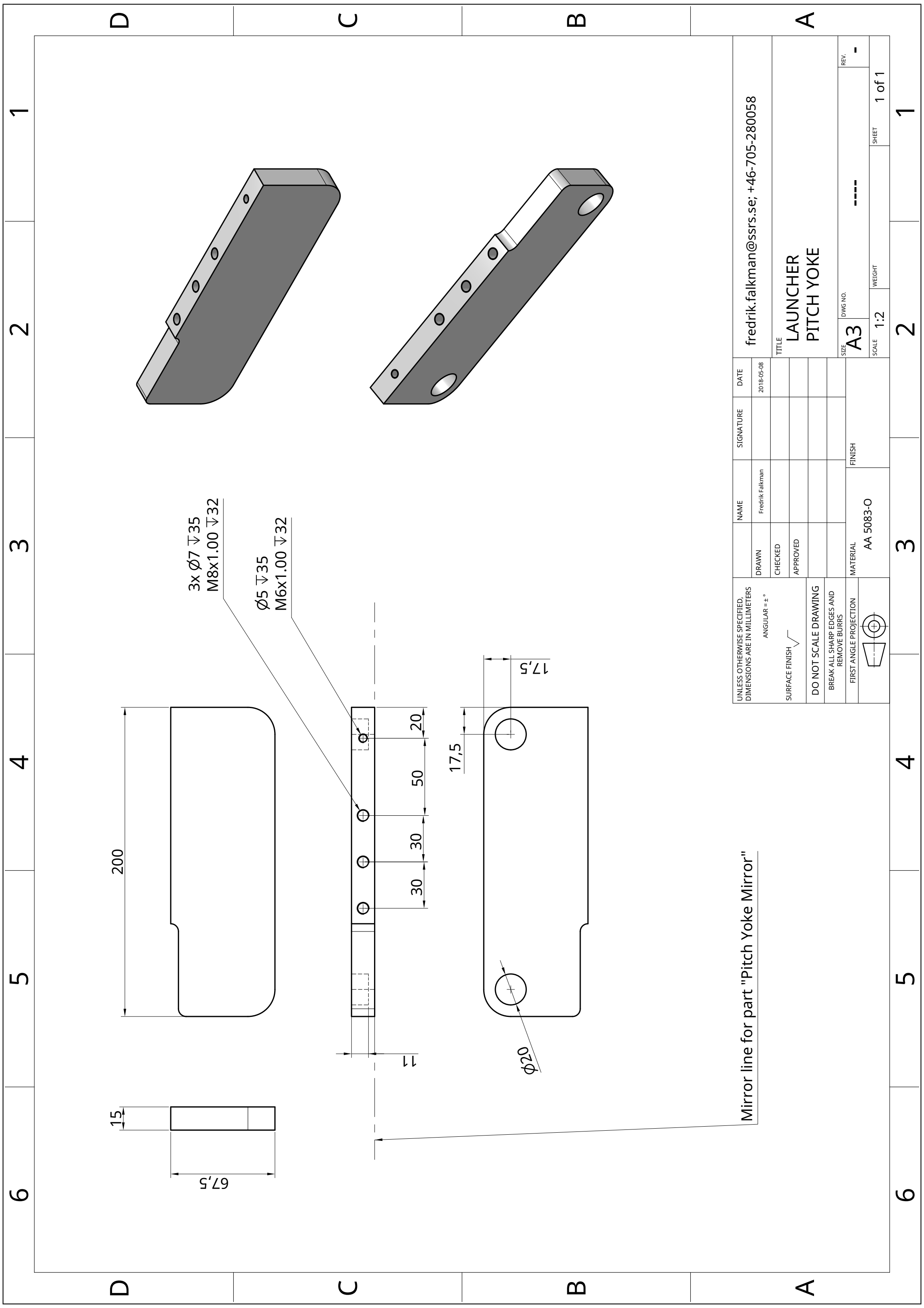


Figure C.9: Pitch yoke.

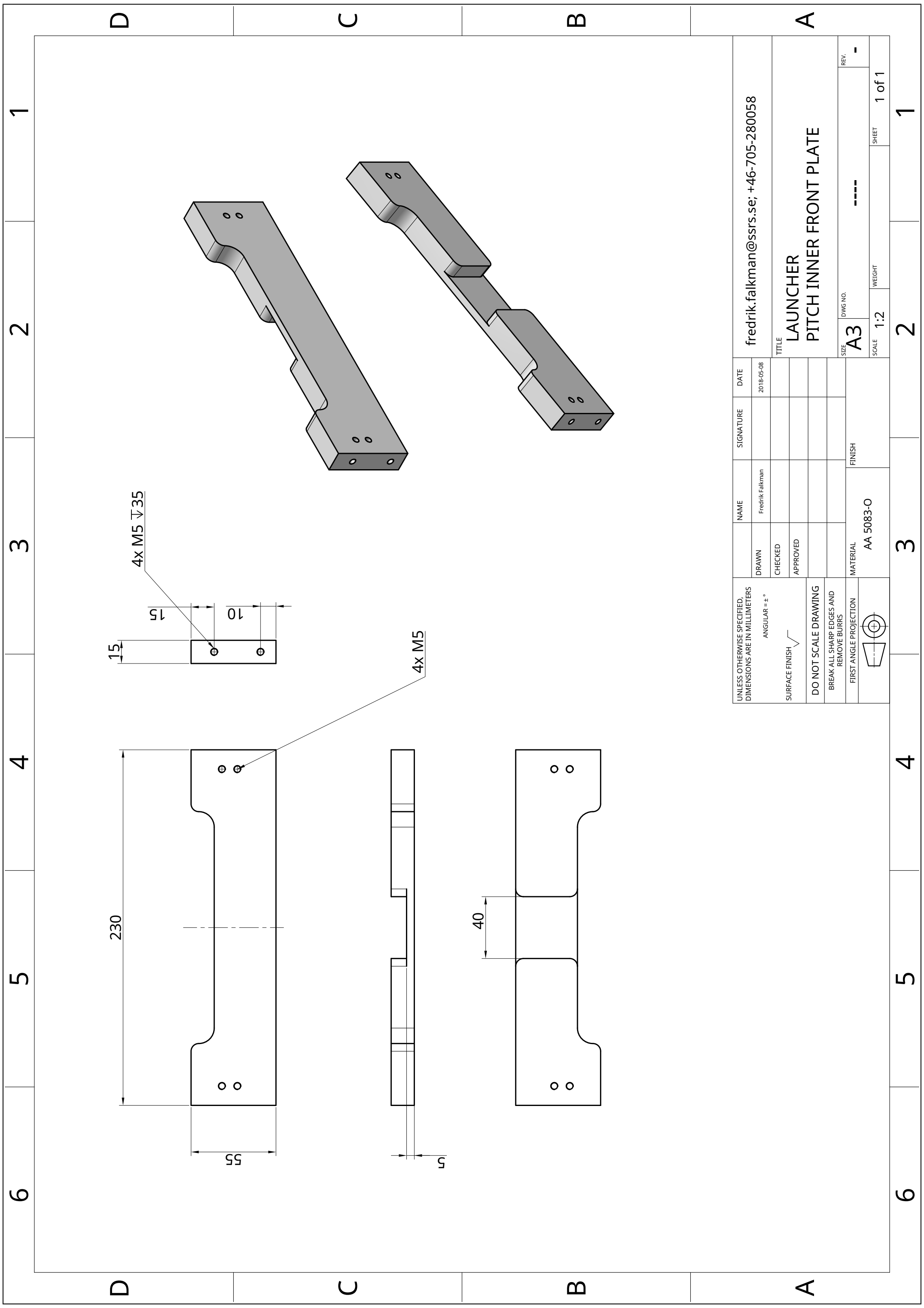


Figure C.10: Pitch inner front plate.

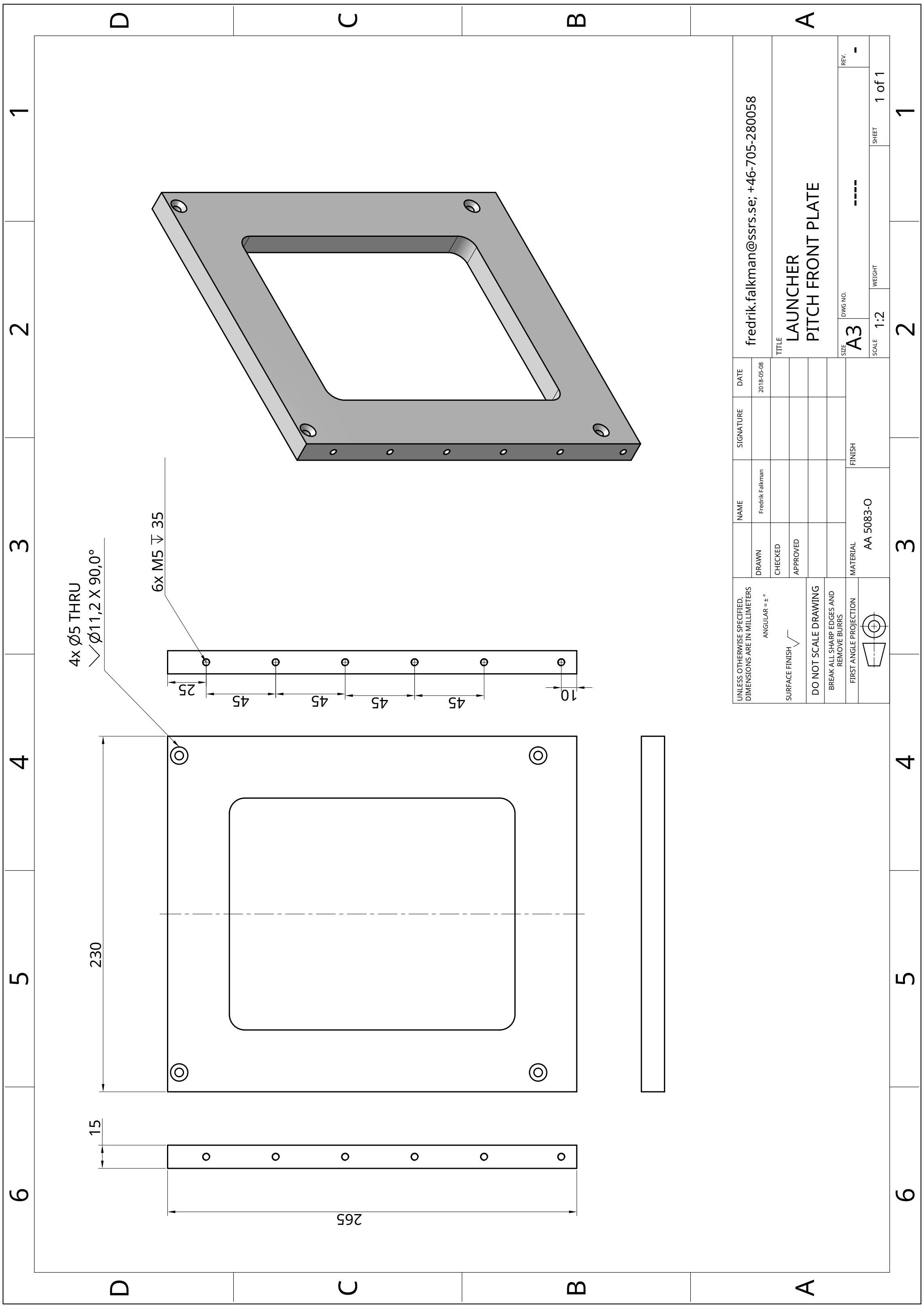


Figure C.11: Pitch front plate.

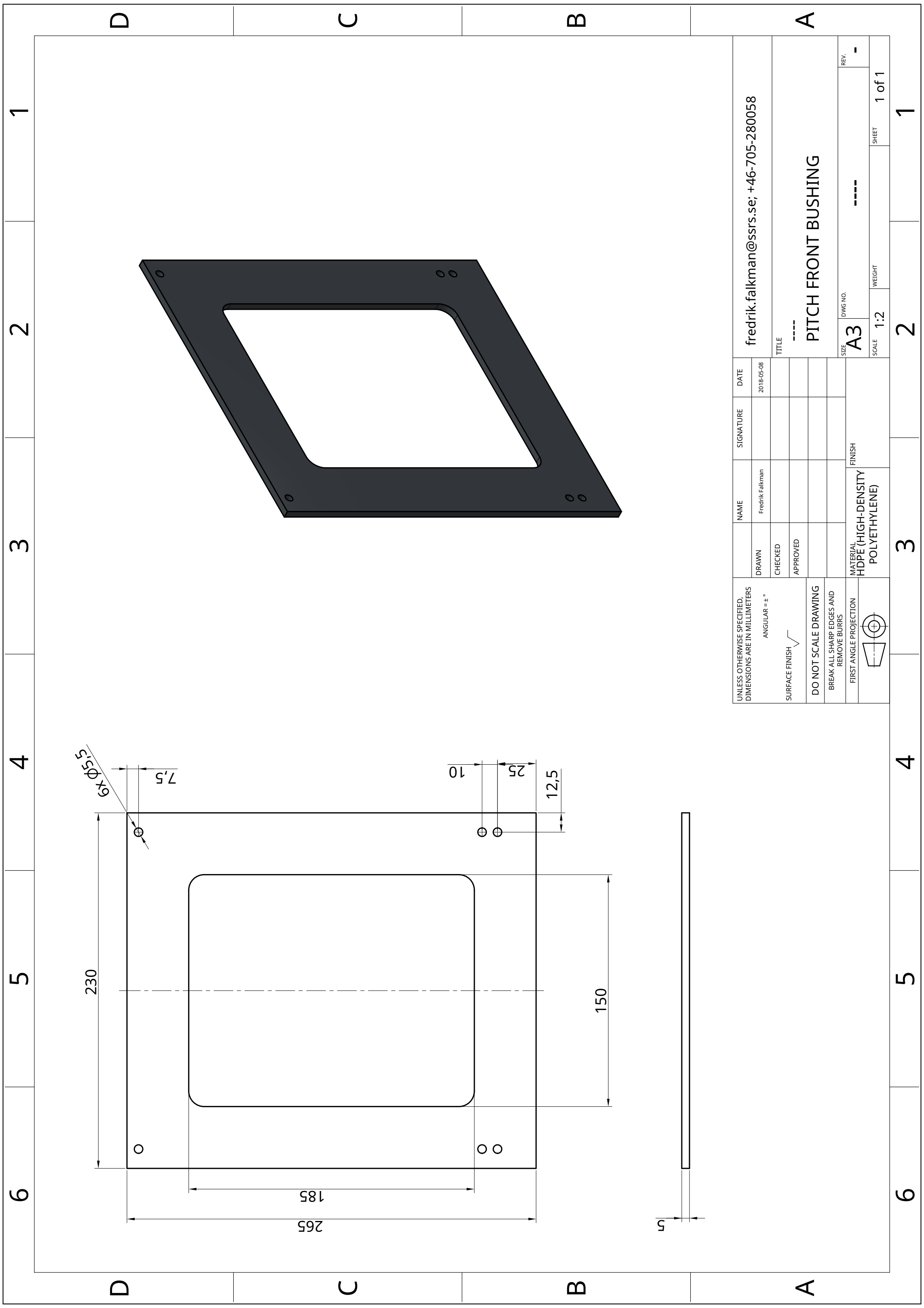


Figure C.12: Pitch front bushing.

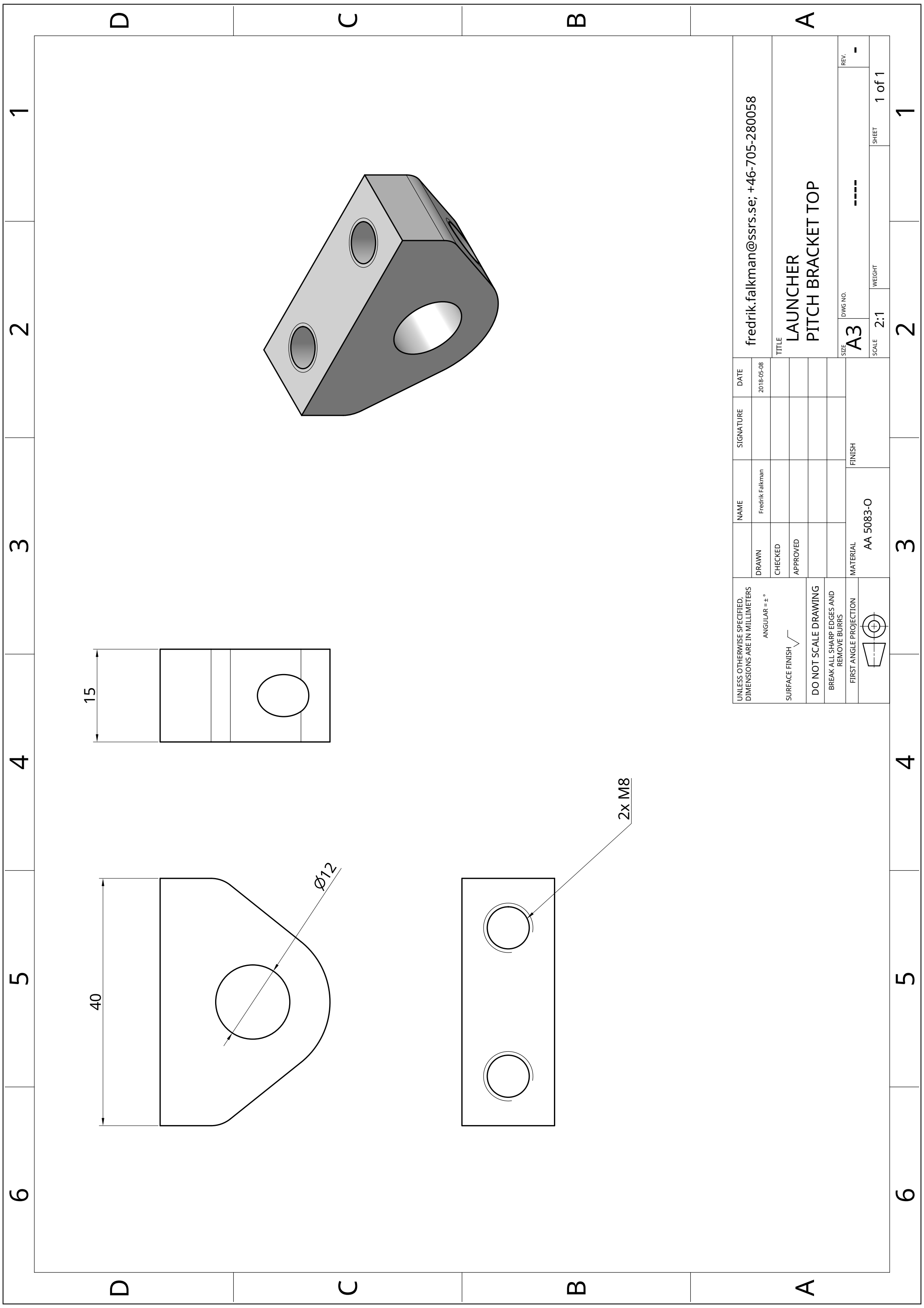


Figure C.13: Pitch bracket top.

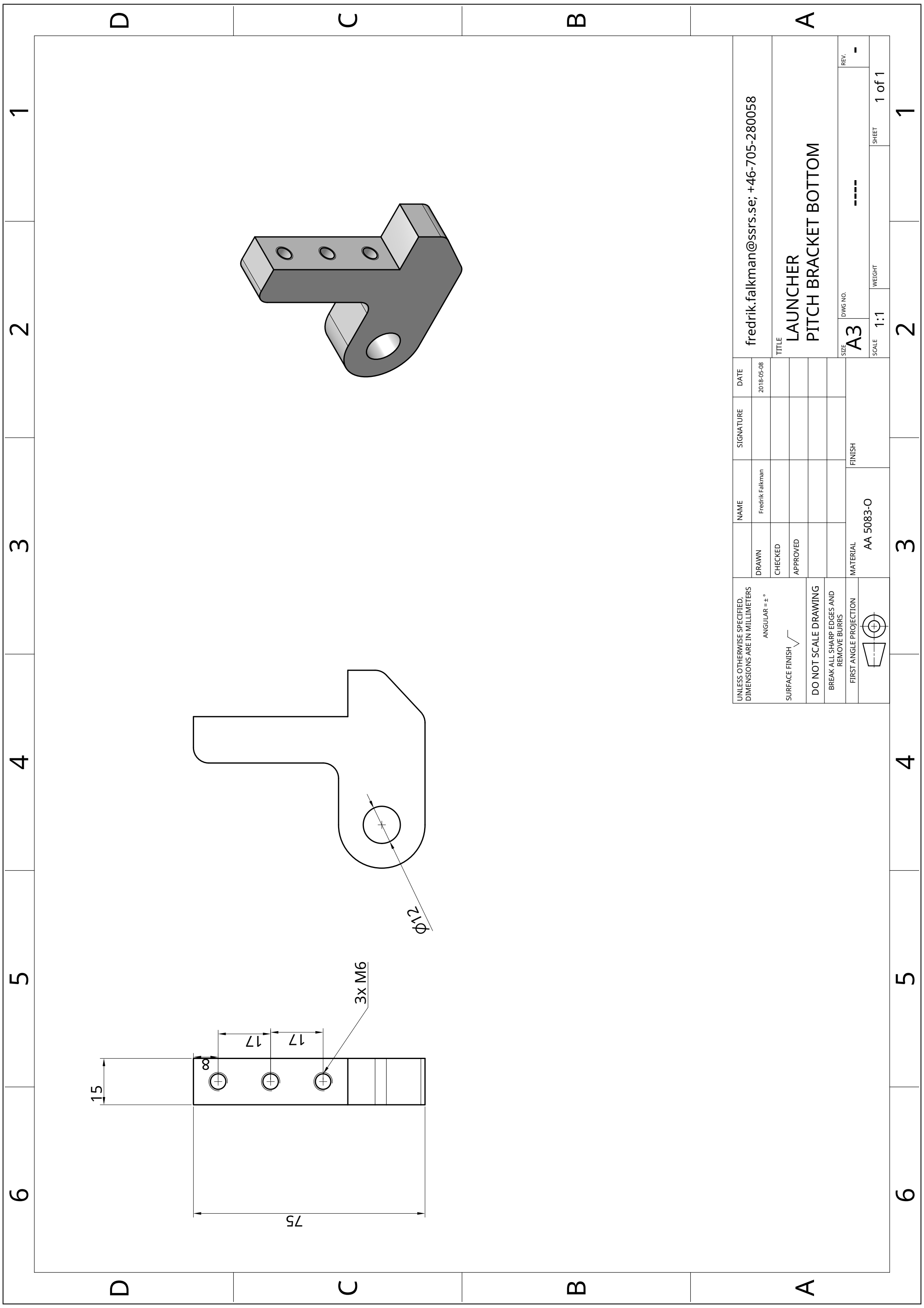


Figure C.14: Pitch bracket bottom.

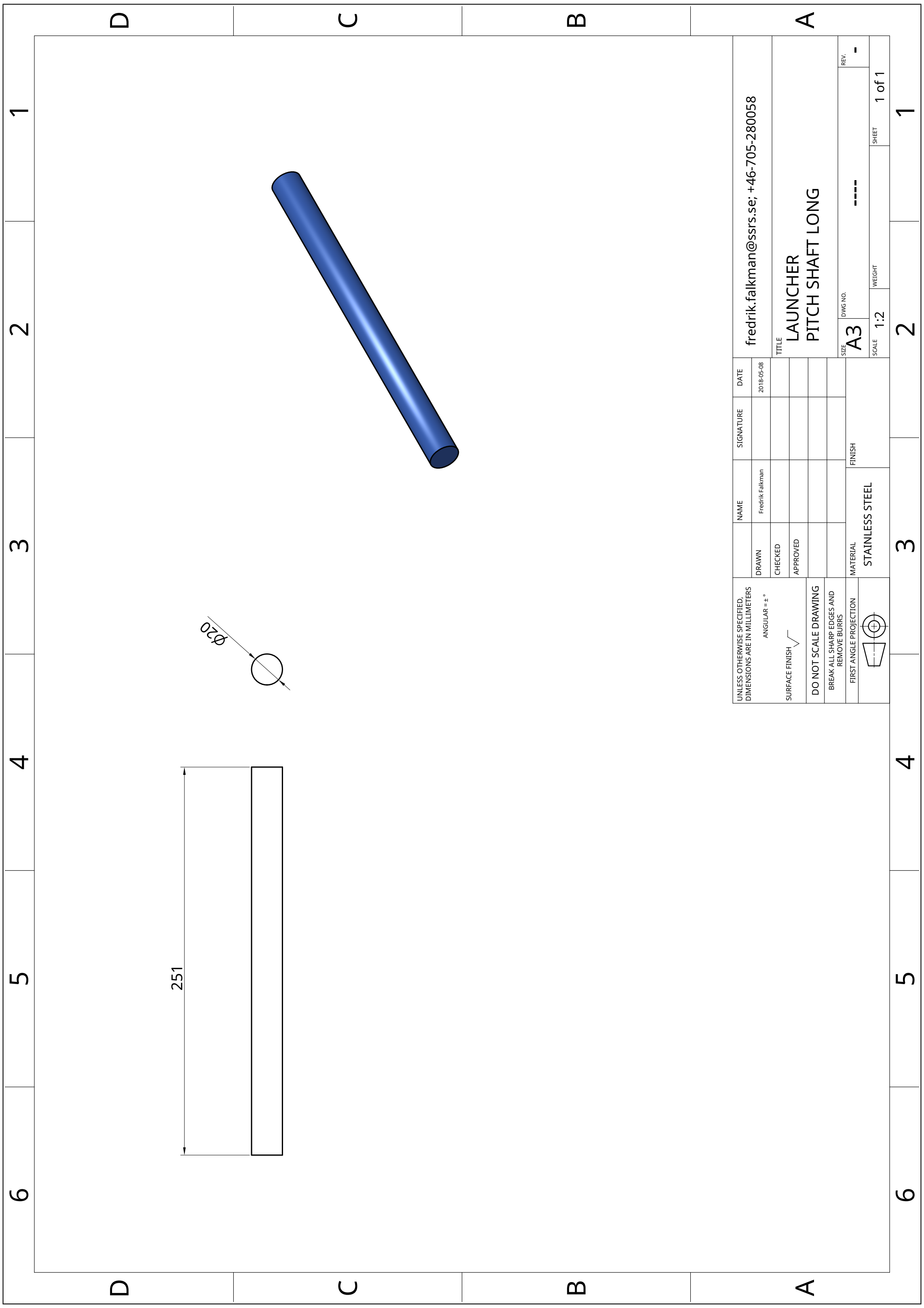


Figure C.15: Pitch shaft long.

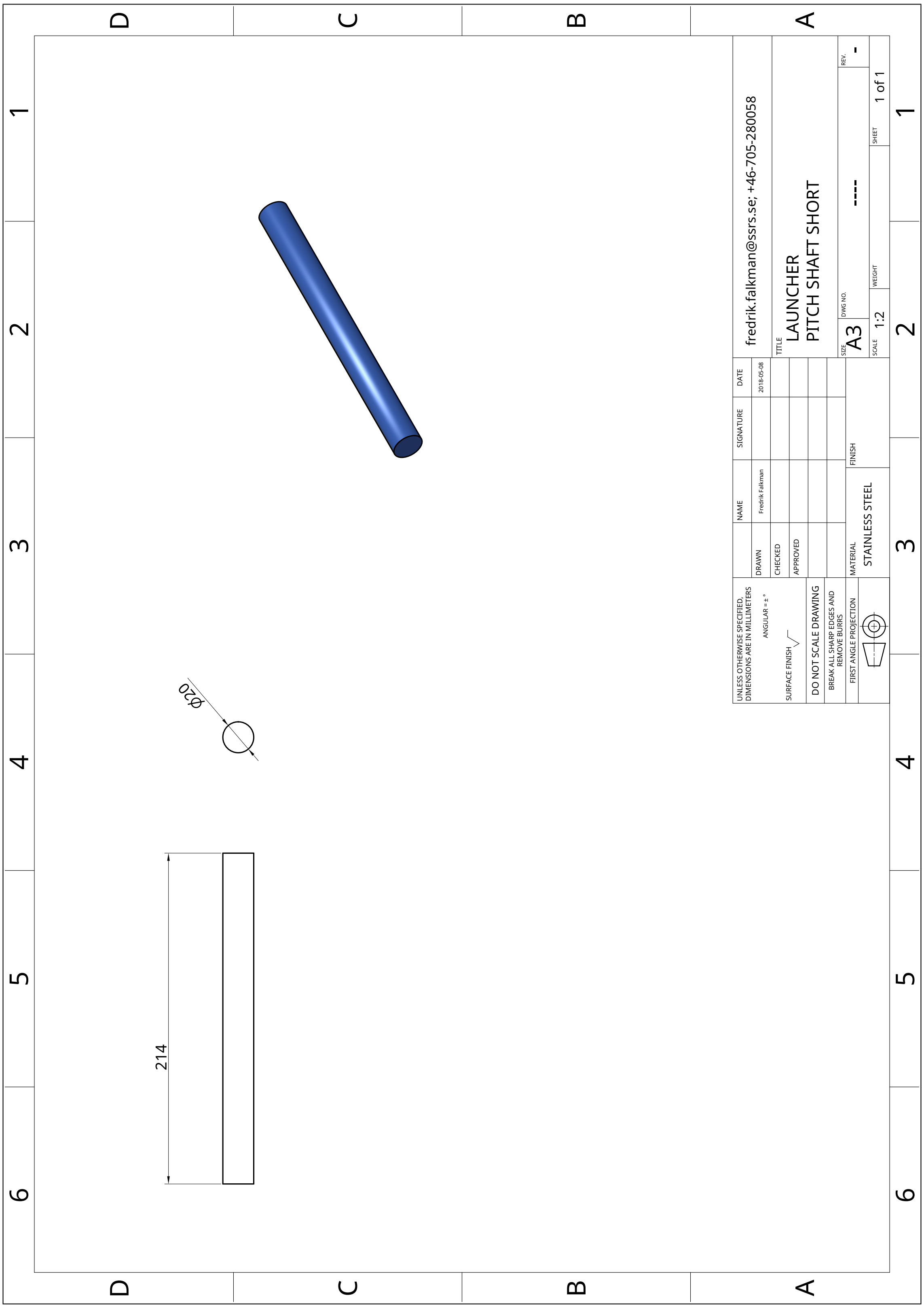


Figure C.16: Pitch shaft short.

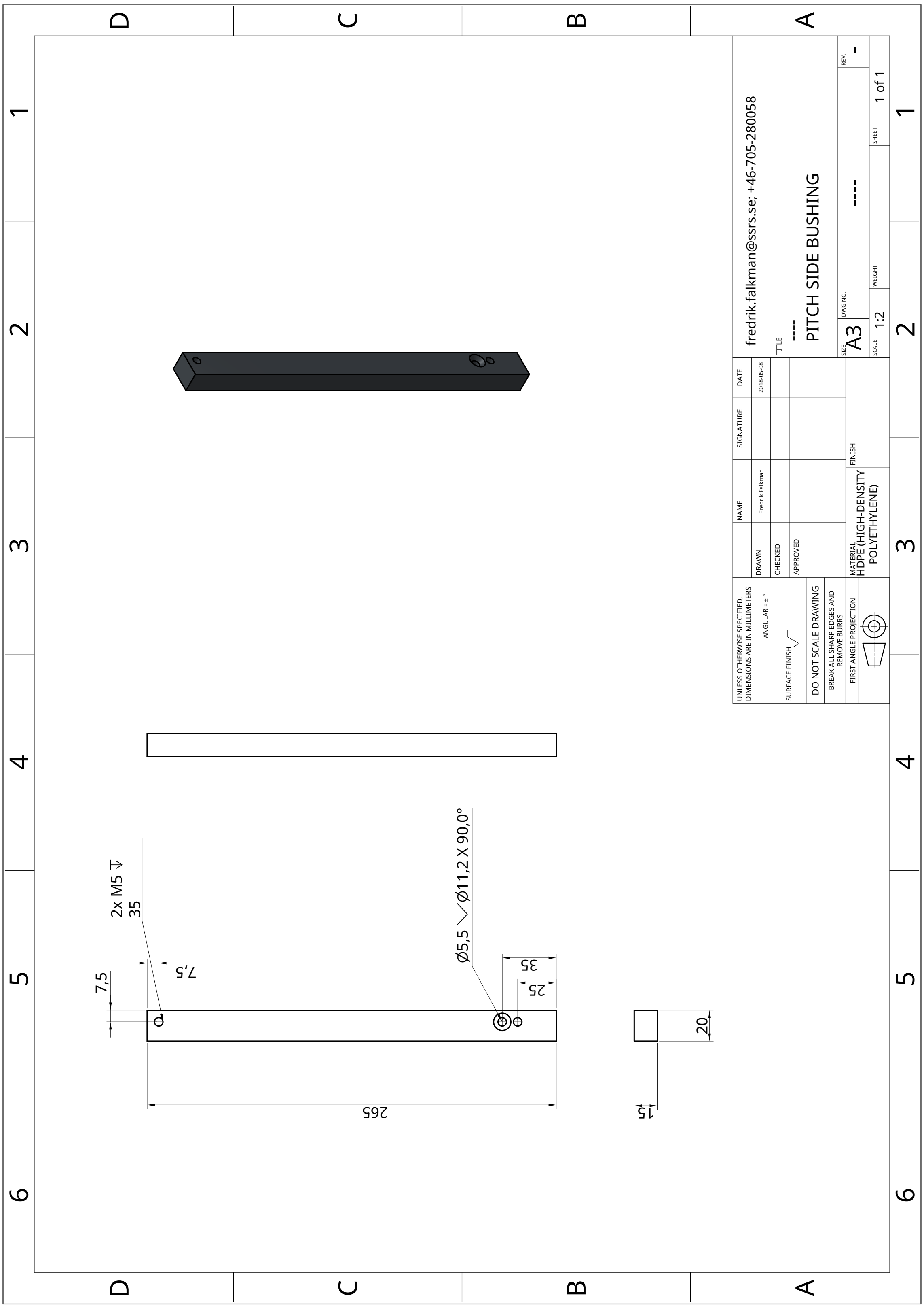


Figure C.17: Pitch side bushing.

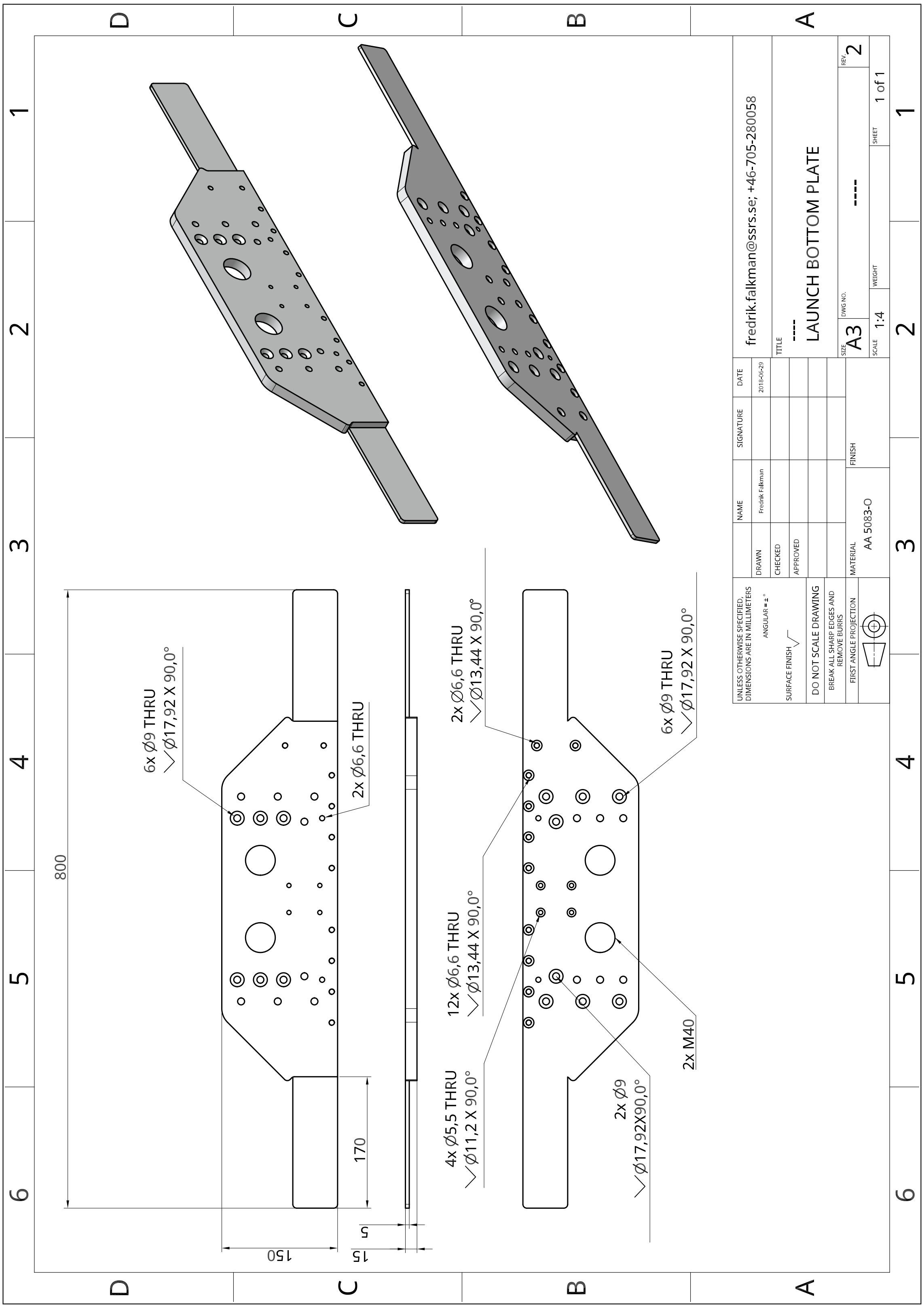


Figure C.18: Launch bottom plate.

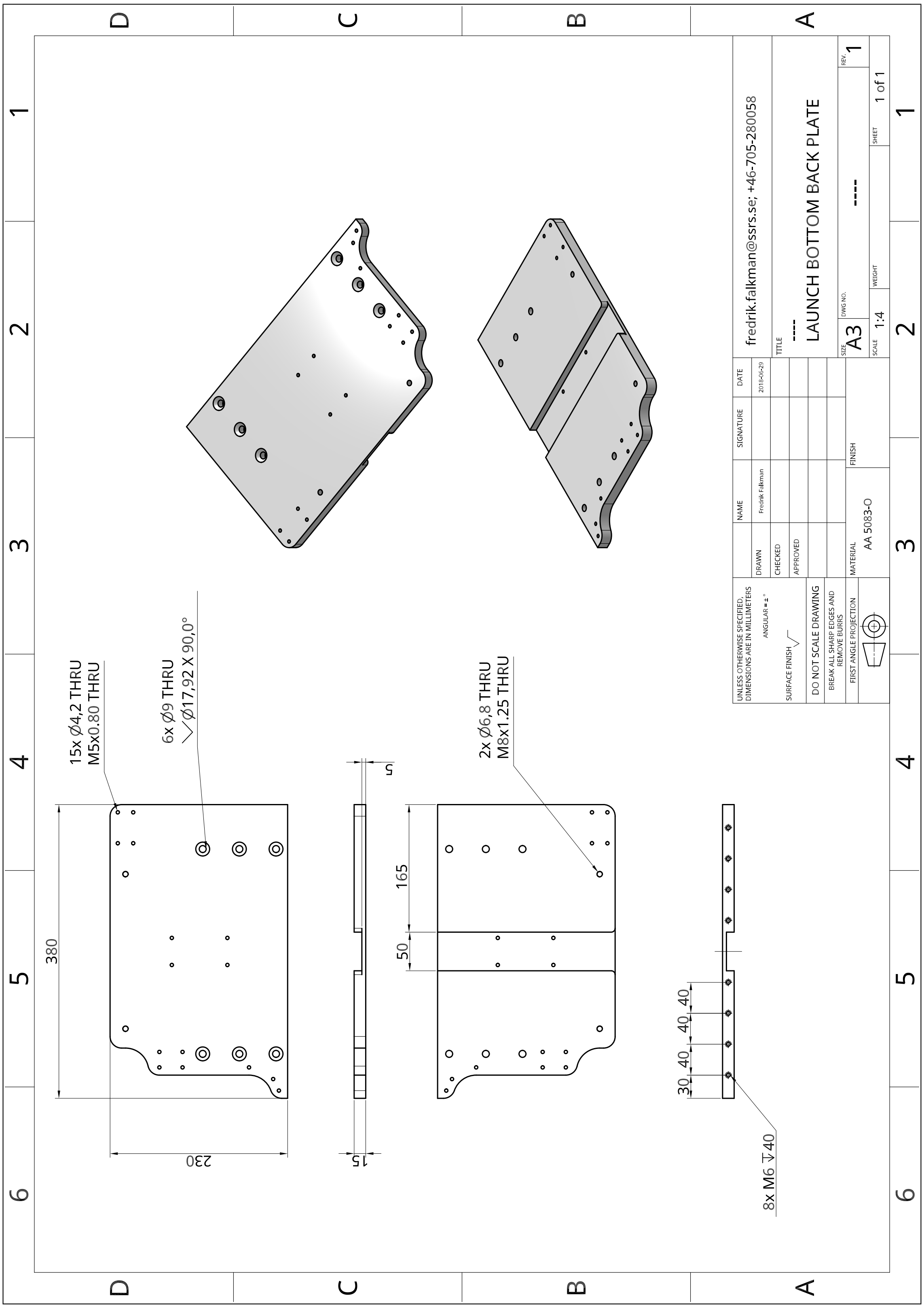


Figure C.19: Launch bottom back plate.

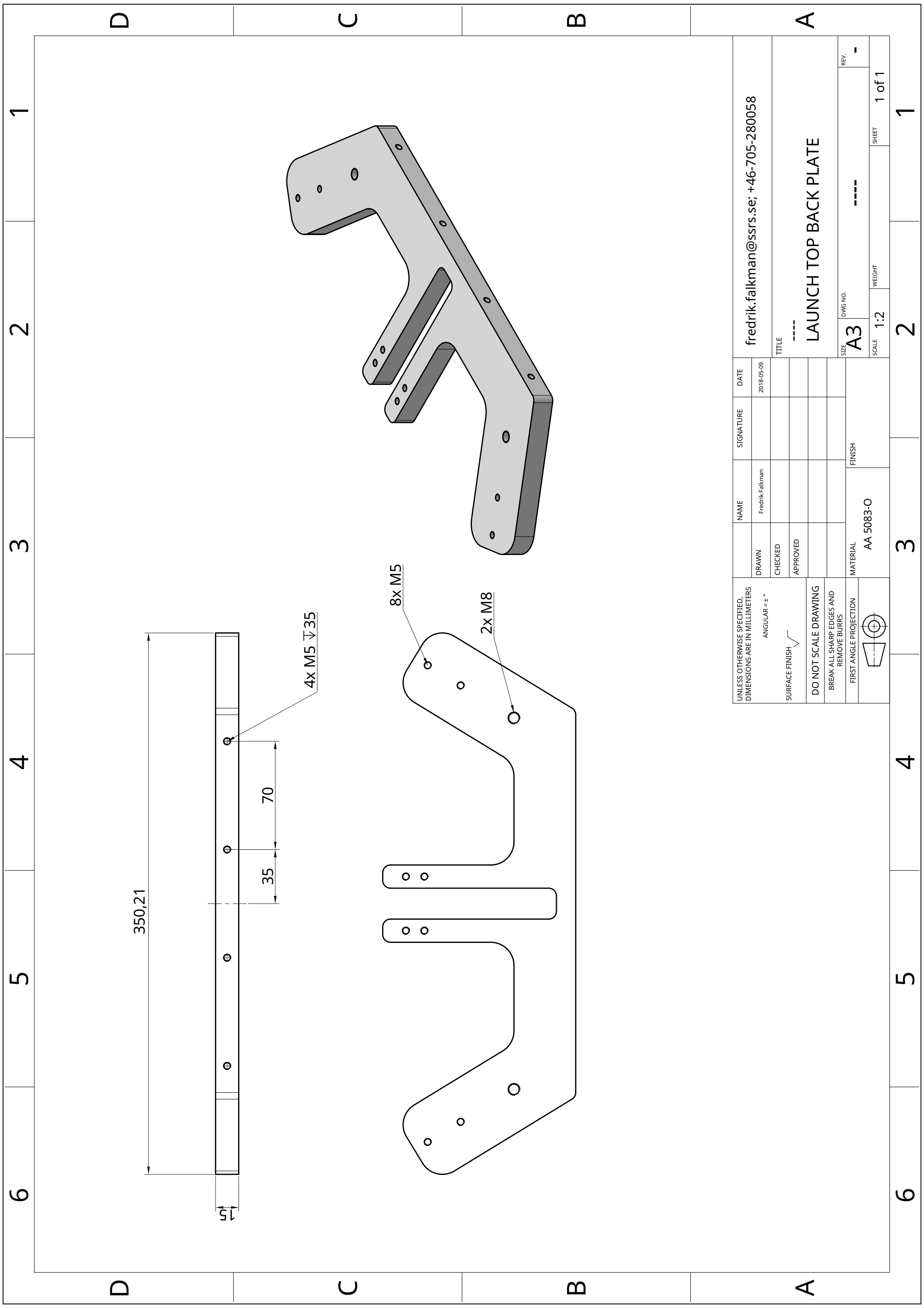


Figure C.20: Launch top back plate.

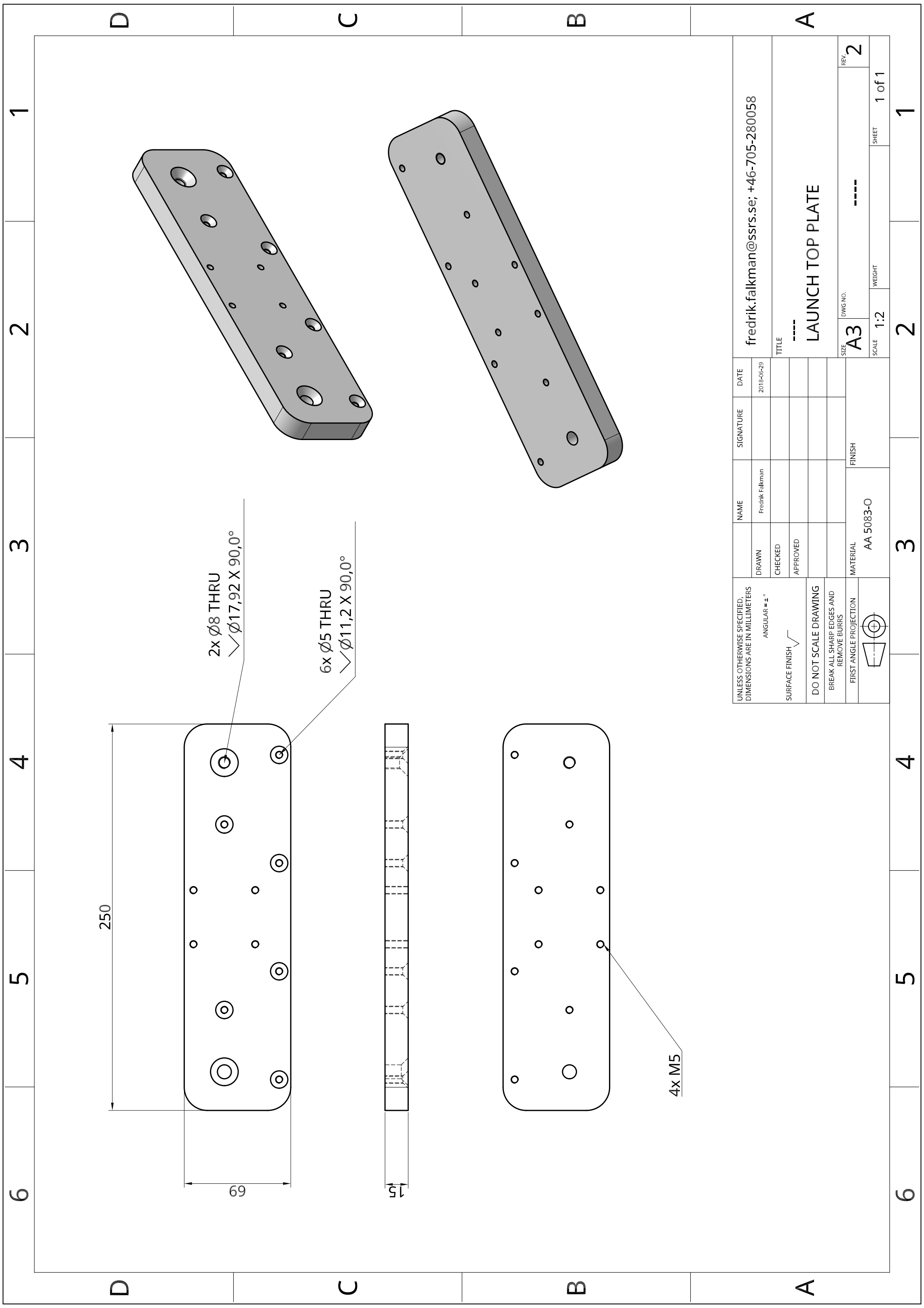


Figure C.21: Launch top plate.

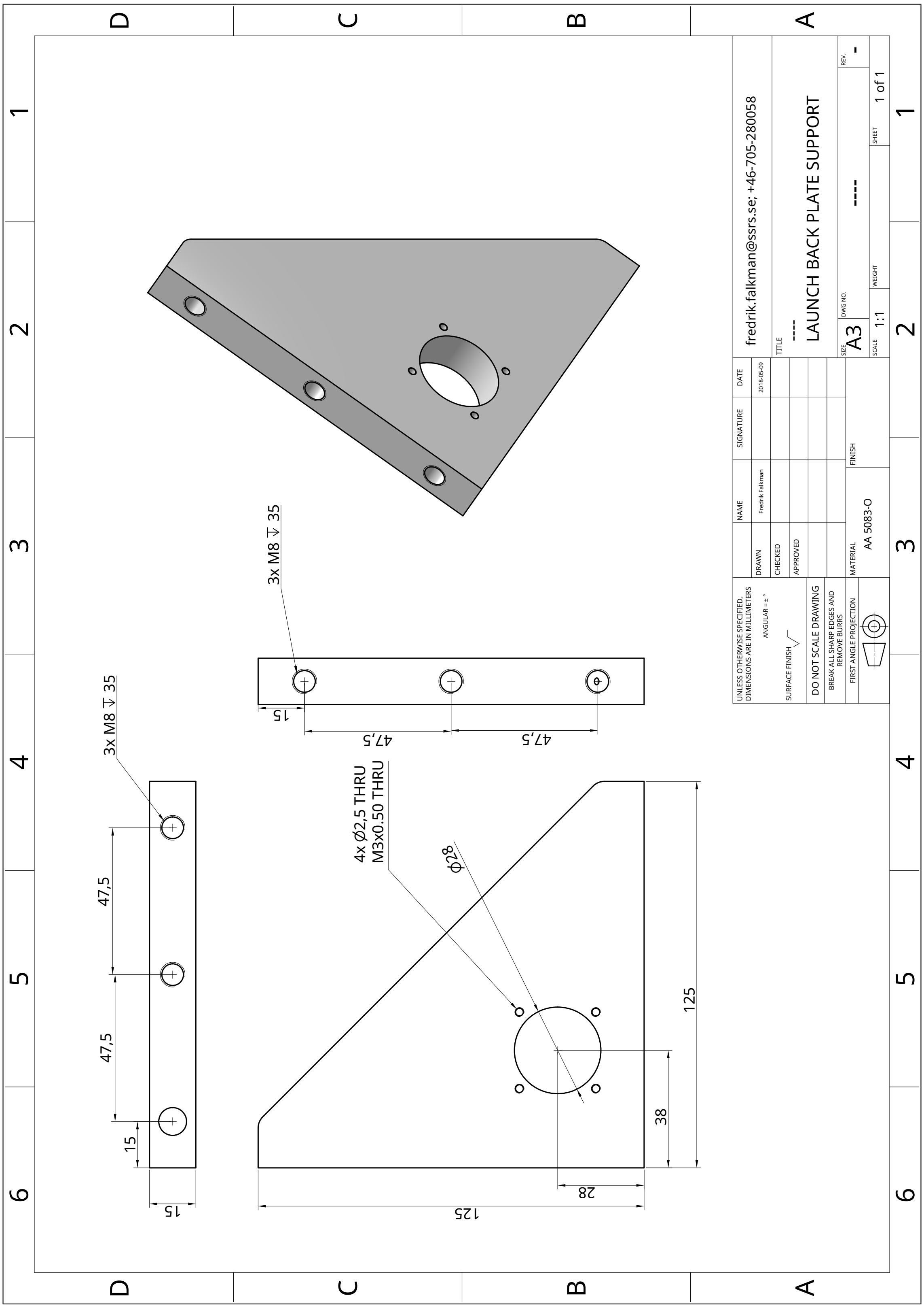


Figure C.22: Launch back plate support.

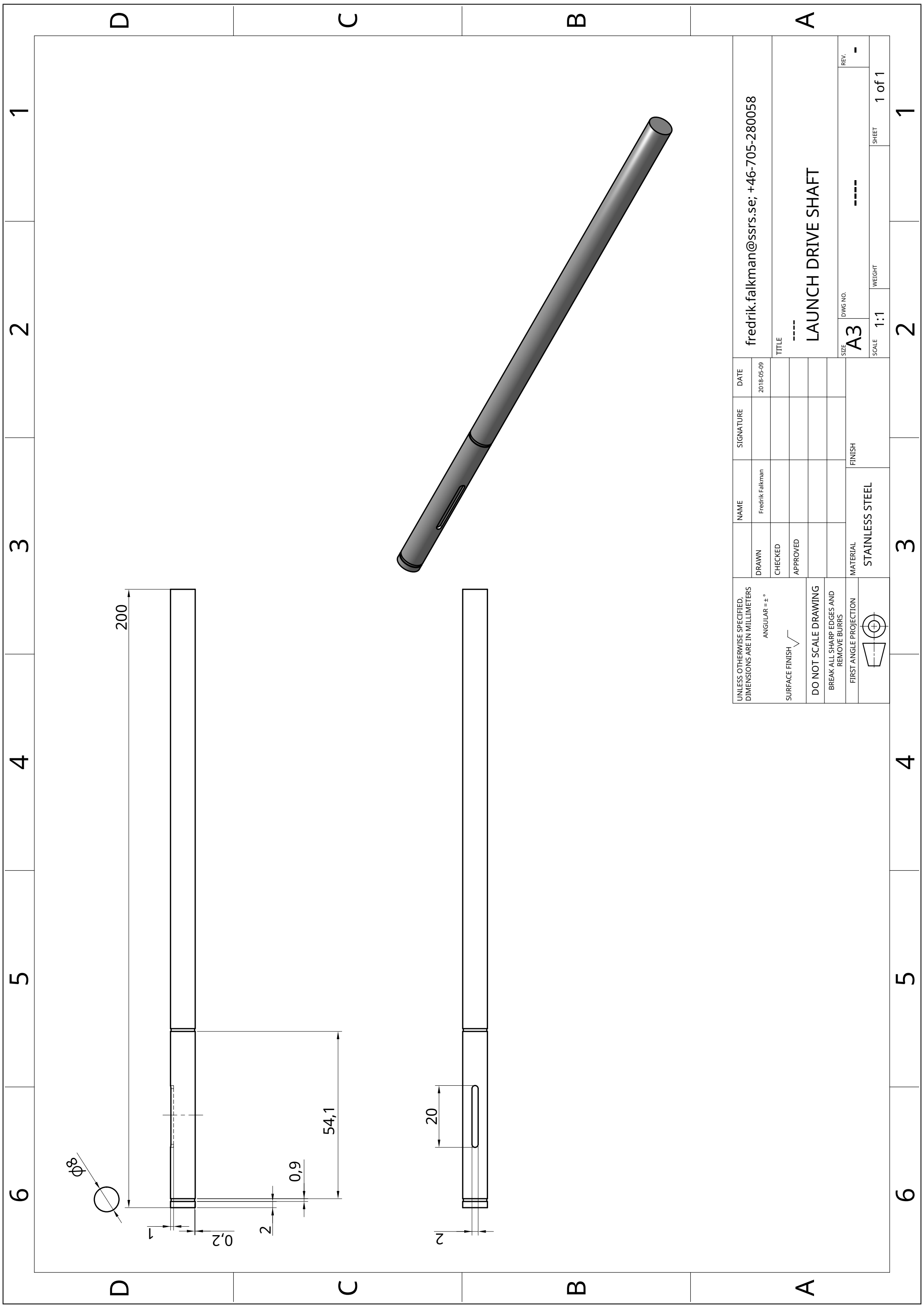


Figure C.23: Launch drive shaft.

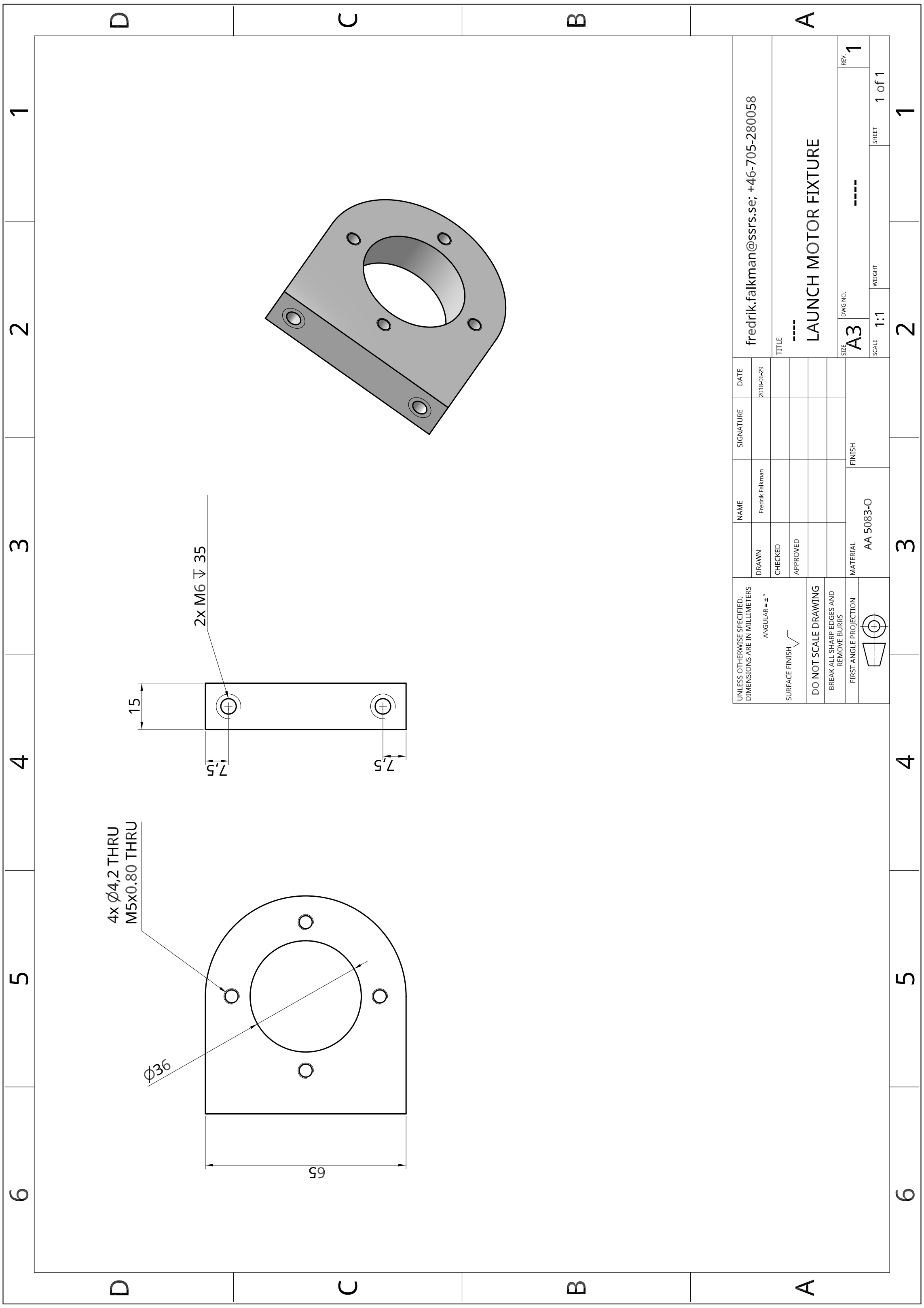


Figure C.24: Launch motor fixture.

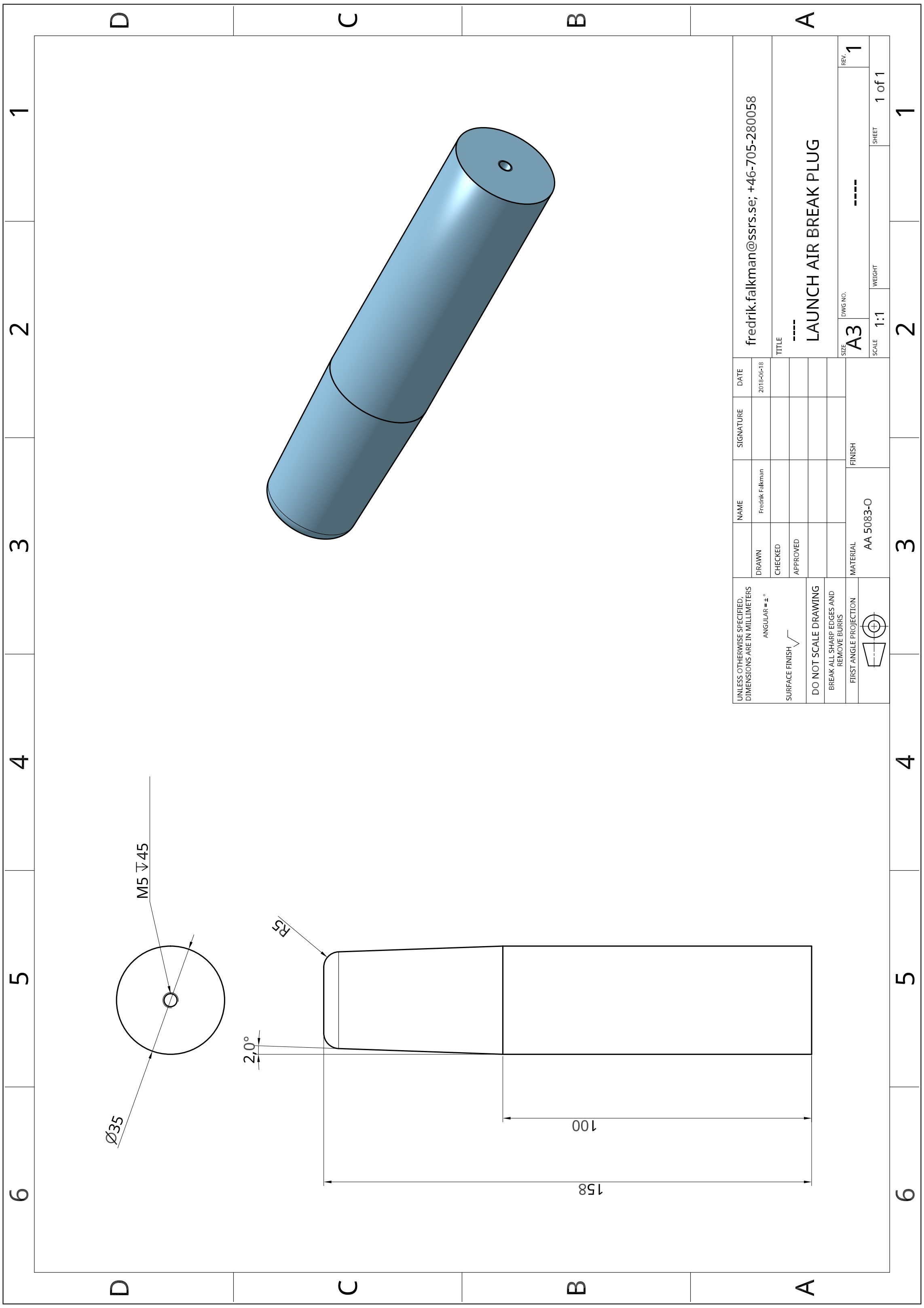


Figure C.25: Launch air break plug.

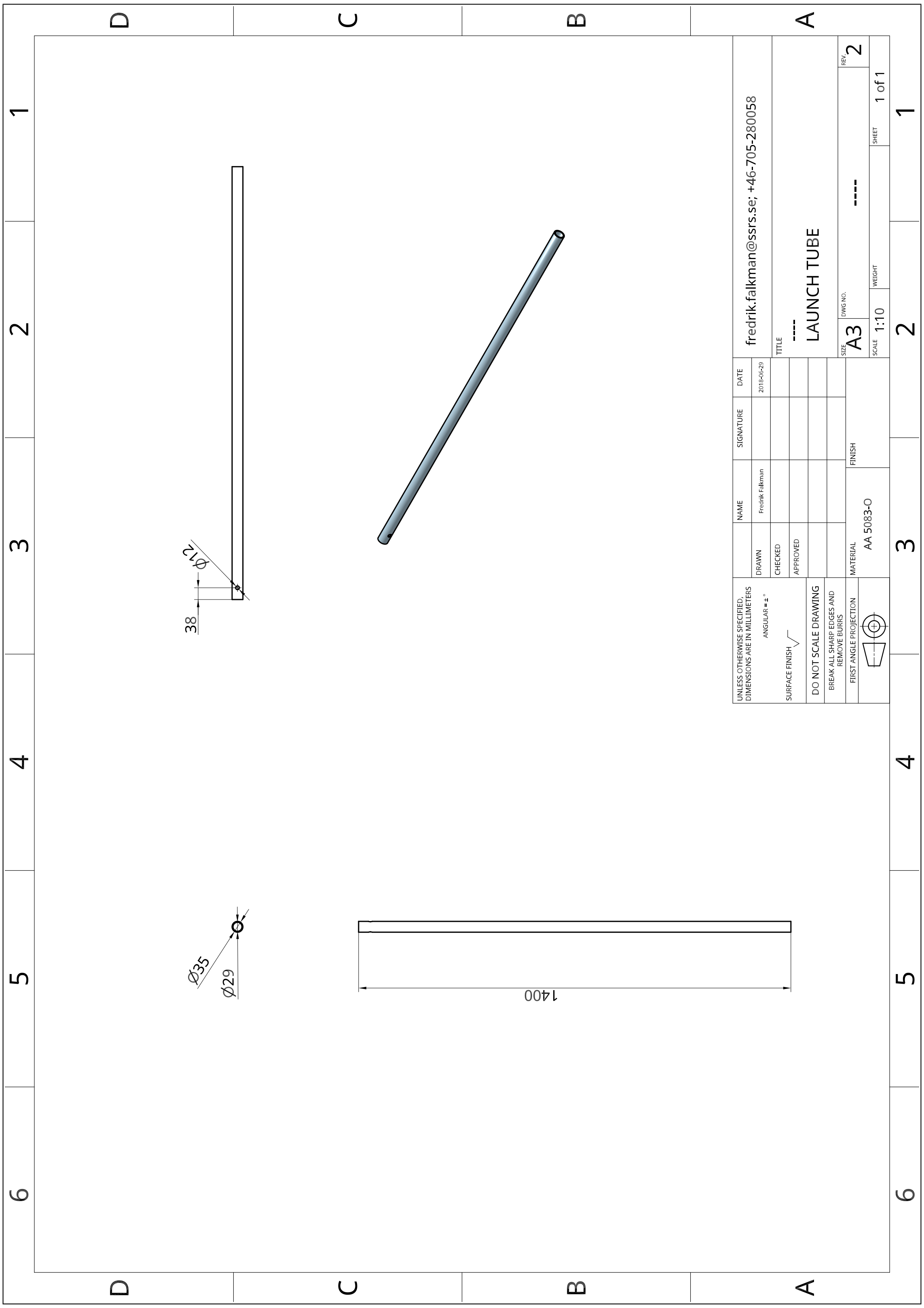


Figure C.26: Launch tube.

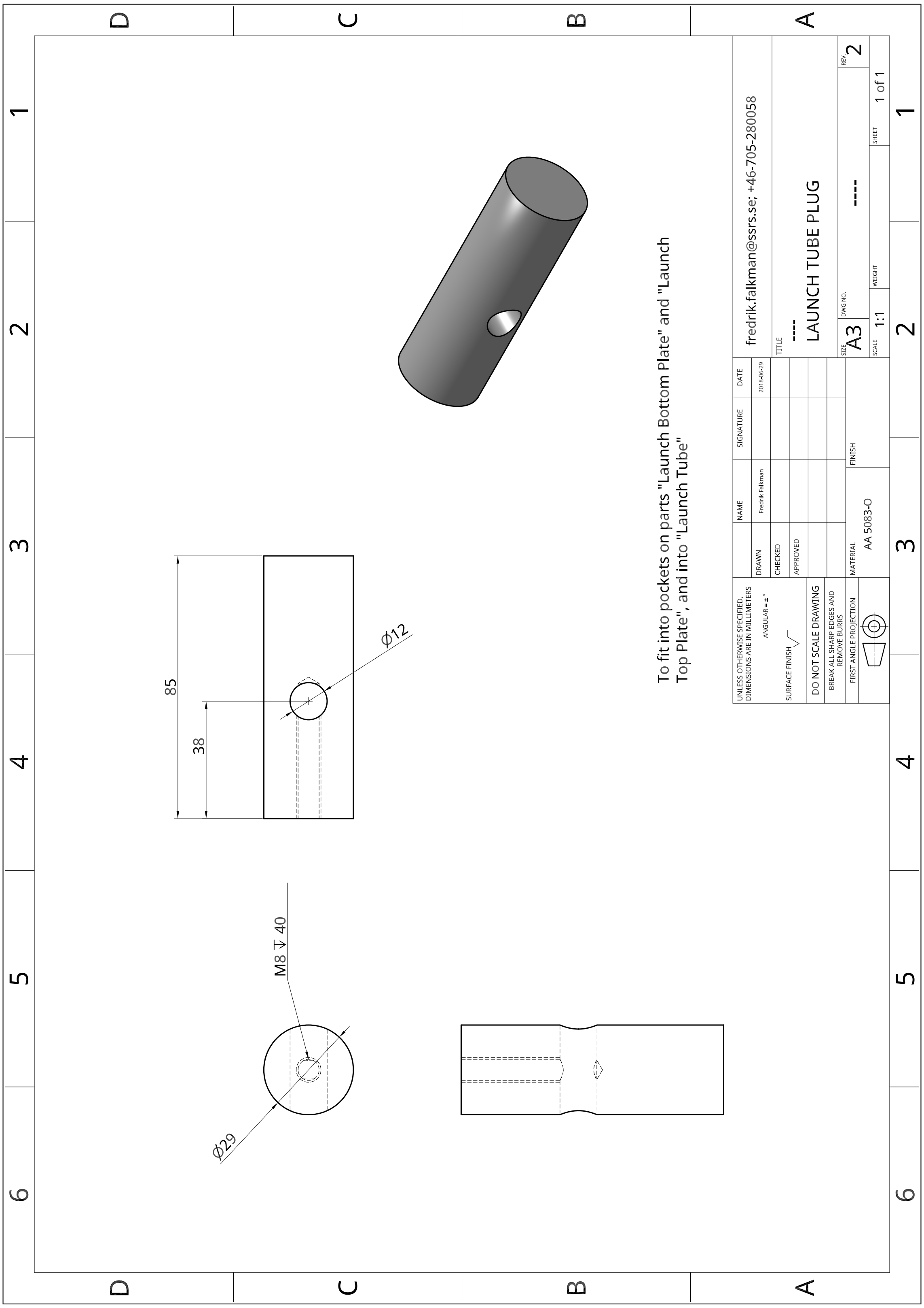


Figure C.27: Launch tube plug.

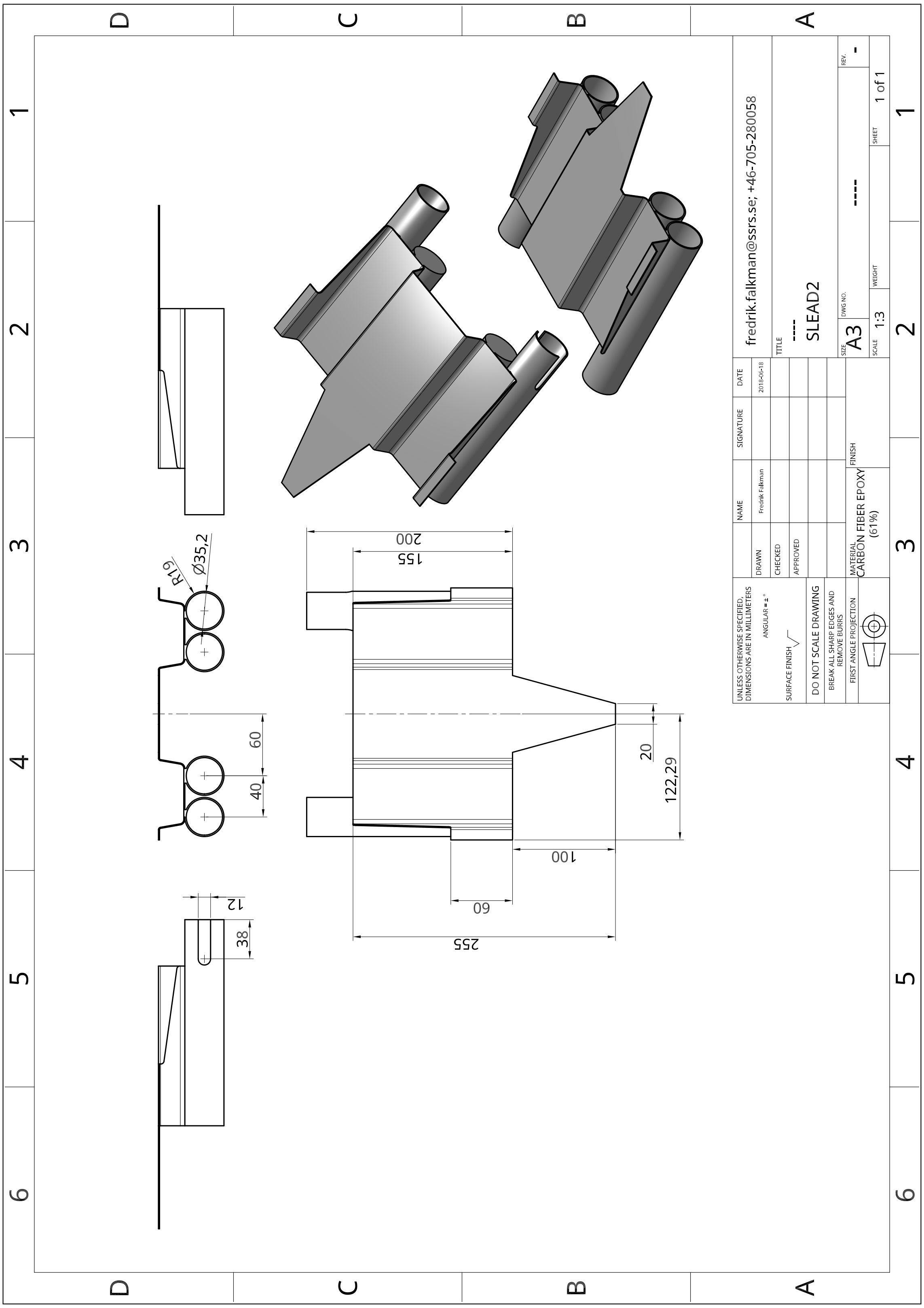


Figure C.28: Slead.

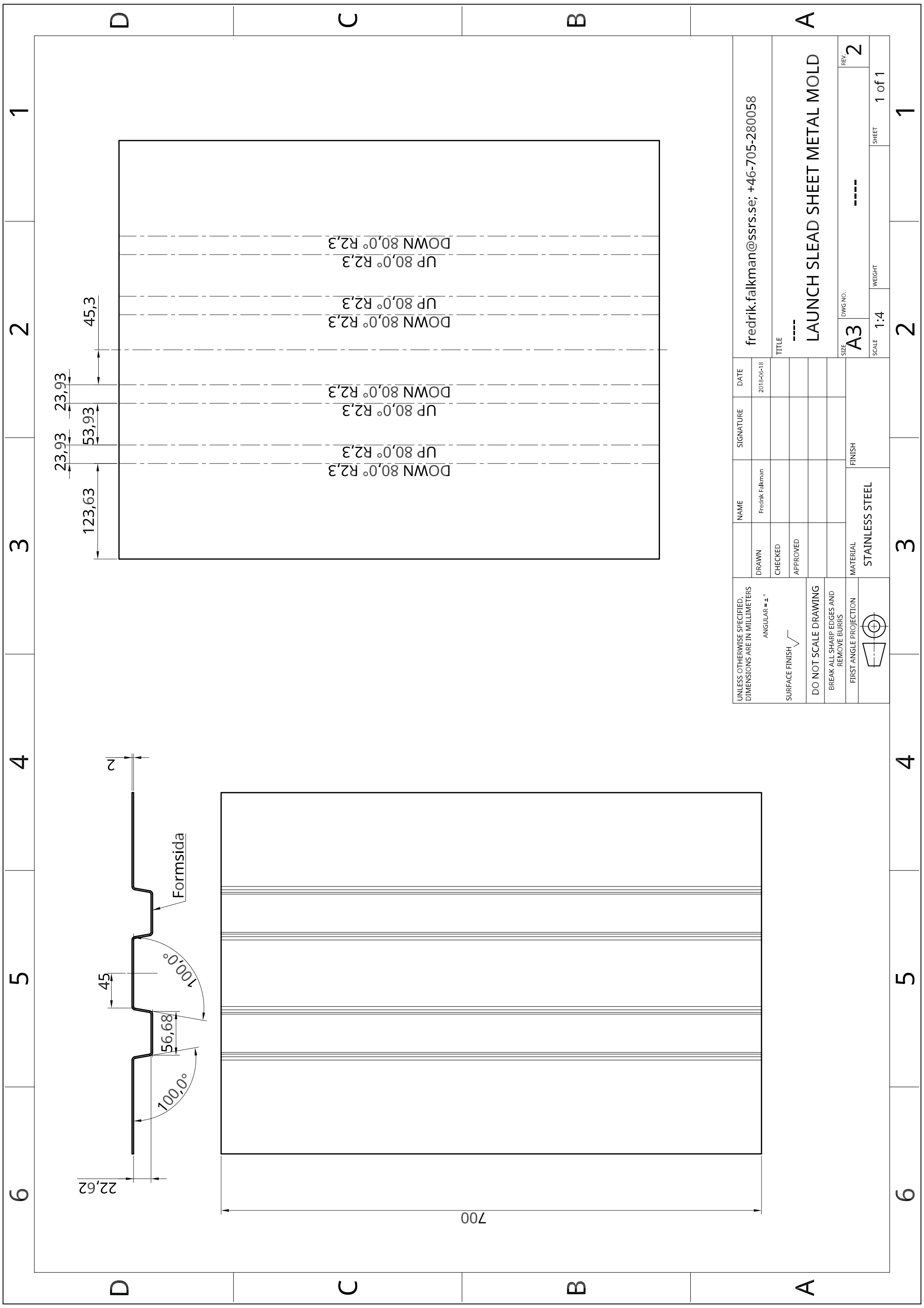


Figure C.29: Slead sheet metal mold.

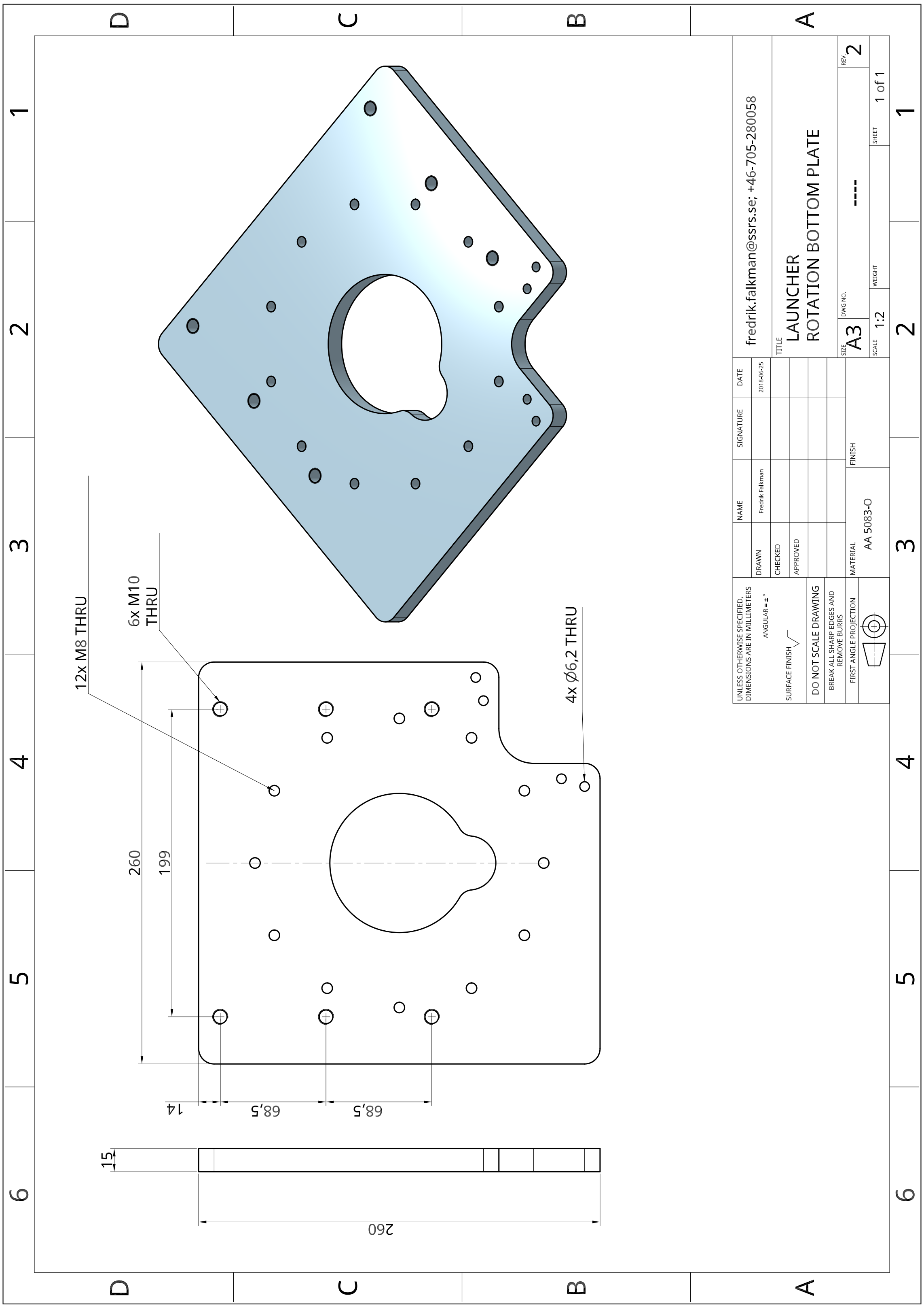


Figure C.30: Rotation bottom plate.

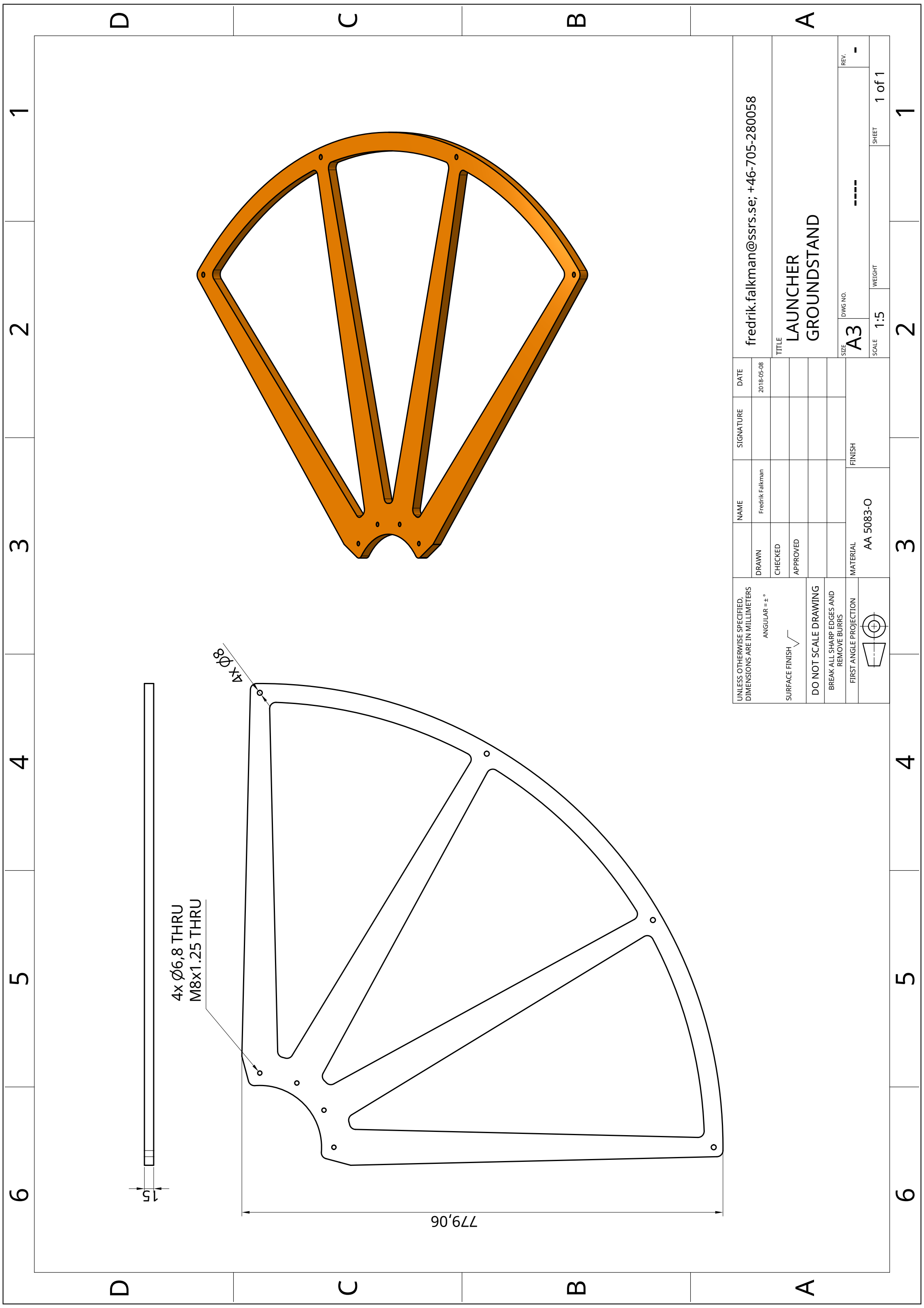


Figure C.31: Rotation groundstand.

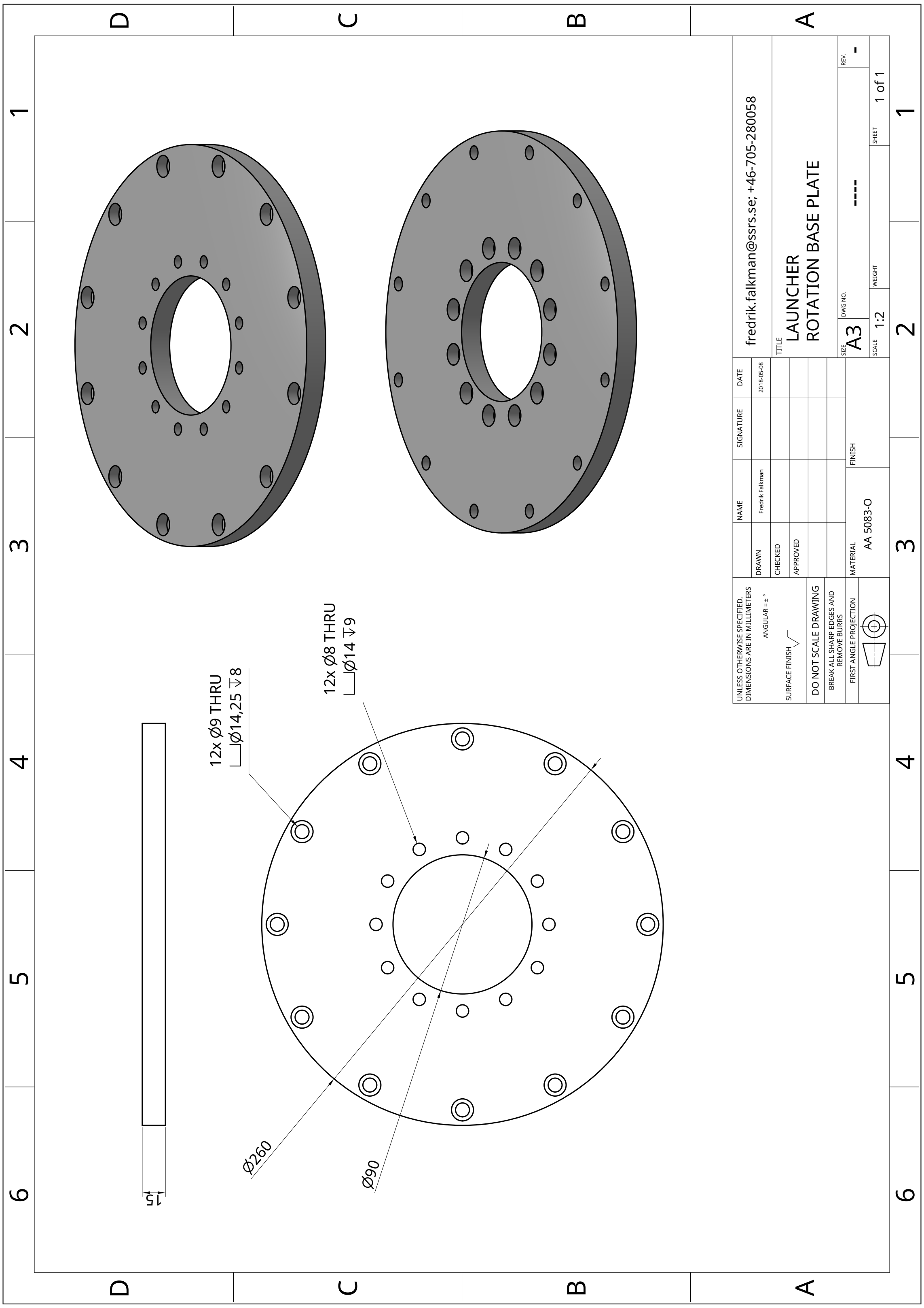


Figure C.32: Rotation base plate.

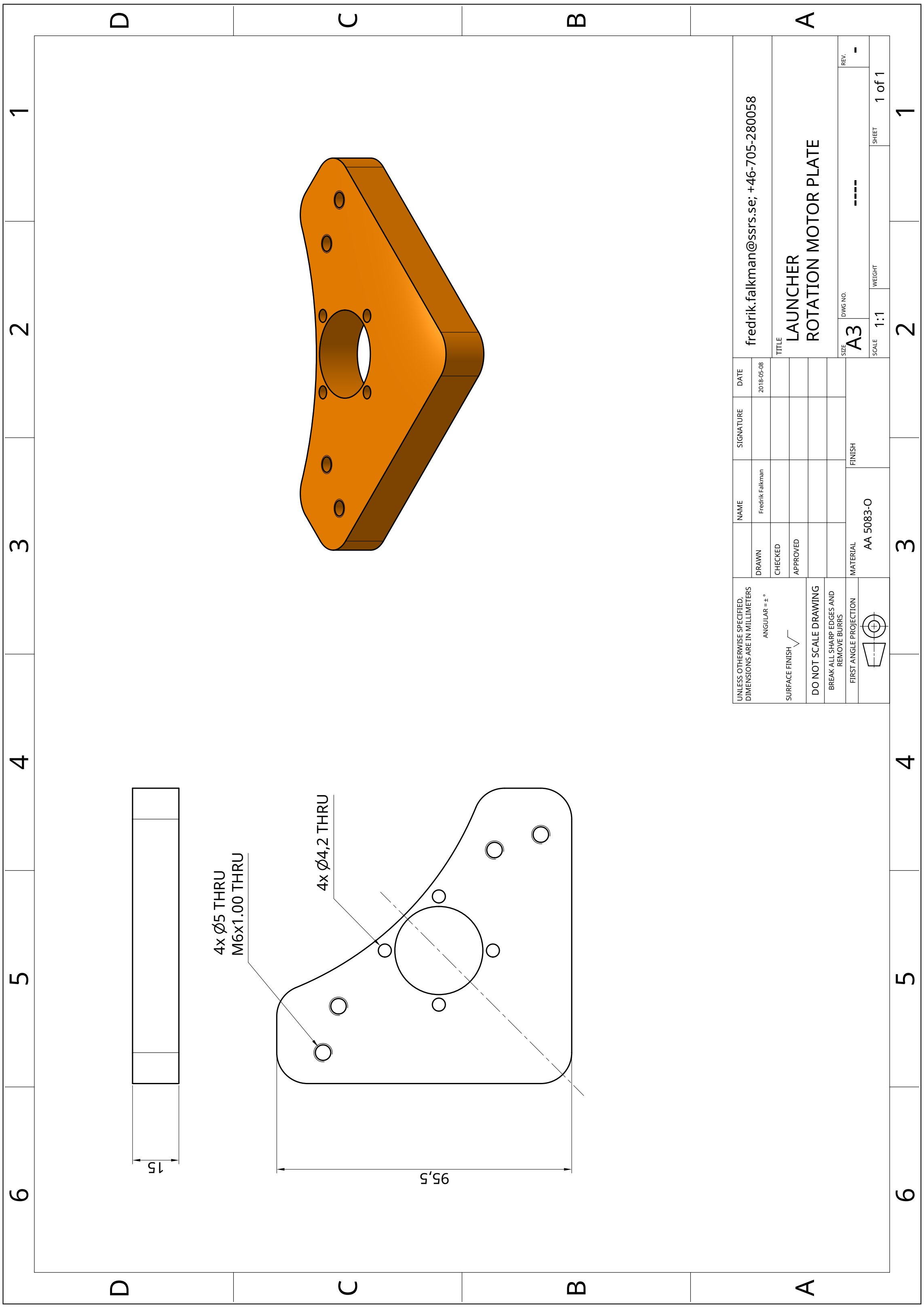


Figure C.33: Rotation motor plate.

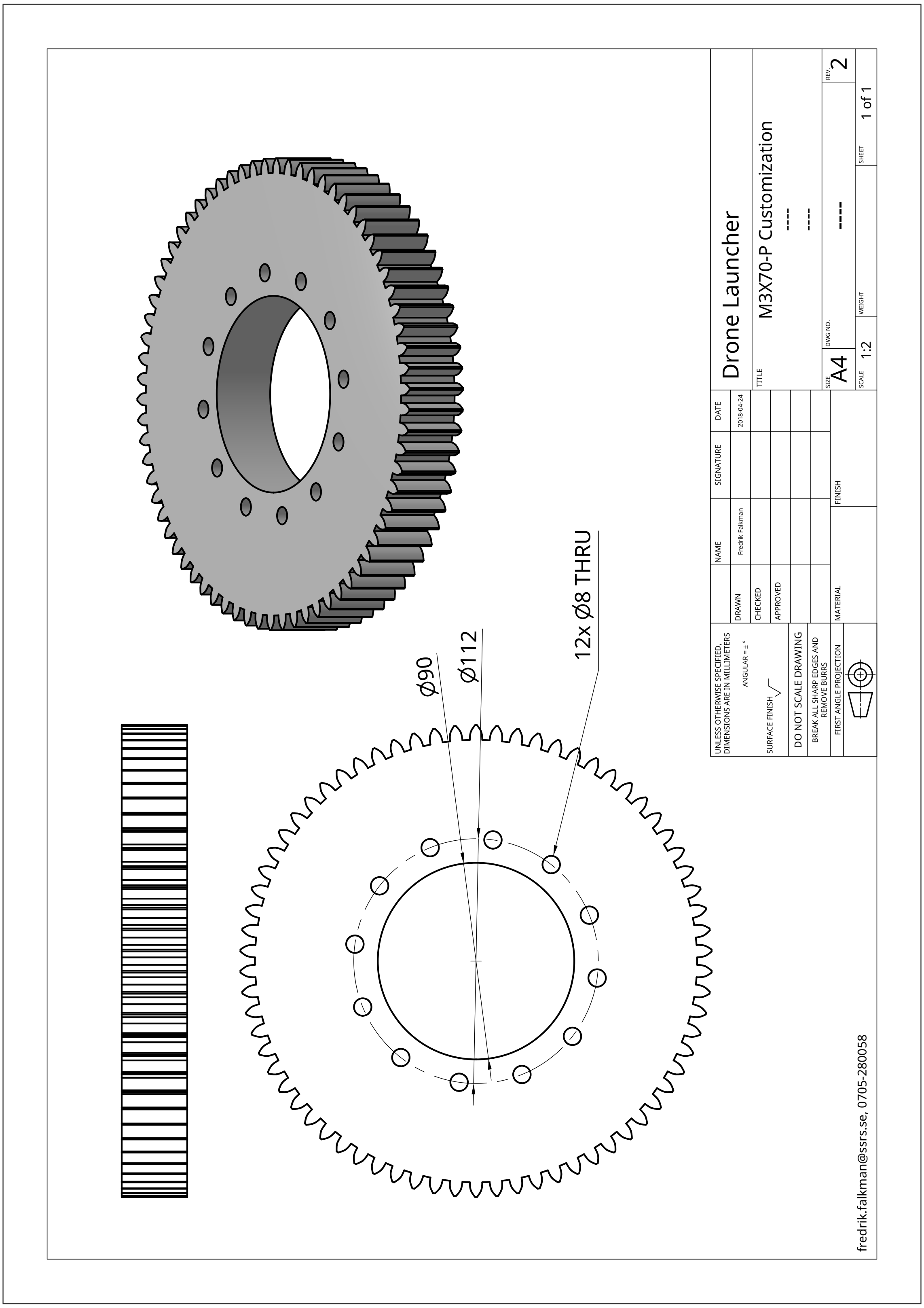


Figure C.34: M3x72-P customization.