

Master Thesis



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Radio Engineering**

Manipulator for an Autonomous Unmanned Aerial Vehicle

Tomáš Kestřánek

Supervisor: Ing. Daniel Heřt

Field of study: Electronics and Communication

Subfield: Technology of the Internet of Things

January 2023

I. Personal and study details

Student's name: **Kestřánek Tomáš** Personal ID number: **456923**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Radioelectronics**
Study program: **Electronics and Communications**
Specialisation: **Technology of the Internet of Things**

II. Master's thesis details

Master's thesis title in English:

Manipulator for an Autonomous Unmanned Aerial Vehicle

Master's thesis title in Czech:

Manipulátor pro bezpilotní autonomní létající dron

Guidelines:

The goal of this thesis is to design and build a lightweight robotic manipulator, with stabilized end effector, which will be attached to an autonomous aerial vehicle (drone).

1. Design and build a robotic manipulator for an unmanned aerial vehicle, using dynamixel servo motors.
2. Design and assemble a printed circuit board which controls the motors of the manipulator, interfaces with the onboard computer of the drone and contains an IMU for stabilizing the manipulator end-effector.
3. Implement an interface for the manipulator and integrate it into the ROS-based [1] MRS UAV System [2]. Implement an algorithm which will stabilize the end effector, to compensate for the tilting of the drone.
4. If the schedule of the MRS group allows, demonstrate the UAV with the manipulator in a real-world experiment.

Bibliography / sources:

[1] Baca et al., "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles", J Intell Robot Syst 102, 26 (2021).

[2] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

Name and workplace of master's thesis supervisor:

Ing. Daniel Heřt Multi-robot Systems FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **24.01.2022** Deadline for master's thesis submission: _____

Assignment valid until: **30.09.2023**


Ing. Daniel Heřt
Supervisor's signature



doc. Ing. Stanislav Vitek, Ph.D.
Head of department's signature


prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

20.12.2022
Date of assignment receipt


Student's signature

Acknowledgements

The biggest thanks go to the author's family who supported him throughout his whole studies and gave him a suitable environment to finish this thesis. Another acknowledgment belongs to this thesis supervisor Ing. Daniel Heřt for appropriate pieces of advice, inspiration, and a generous amount of patience.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 10. January 2023

Abstract

After a short introduction, the thesis describes the process behind designing and building a lightweight stabilized robotic manipulator for a UAV. In its first part, the thesis focuses on the hardware design which is about the mechanical construction of the manipulator itself and the design of custom electronics together with a PCB. The second part is focused on software where a control algorithm that collects data from an onboard IMU and stabilizes the end effector is implemented on an STM32F411RE. The software part also includes communication with the UAV's computer running ROS via UART. At the end of the thesis, the solution is tested on real hardware and a conclusion is made.

Keywords: uav, autonomous, unmanned, aerial, vehicle, manipulator, stabilization, STM32, IMU, drone

Supervisor: Ing. Daniel Heřt

Abstrakt

Po krátkém úvodu práce popisuje proces designu a sestavení lehkého stabilizovaného robotického manipulátoru pro bezpilotní letoun. Ve své první části je práce zaměřená na návrh hardwaru. Ukazuje jak mechanický design samotného manipulátoru, tak návrh elektroniky a DPS potřebných pro pohyb manipulátoru. Druhá část práce se věnuje softwaru a popisuje návrh řídicího algoritmu, který sbírá data z palubní jednotky IMU a stabilizuje chapadlo manipulátoru vzhledem k náklonům letounu. Dále se práce věnuje implementaci algoritmu na procesoru STM32F411RE a komunikaci s počítačem na UAV, na kterém je spuštěn ROS prostřednictvím UART protokolu. Na konci práce je popsané řešení otestováno na reálném hardwaru.

Klíčová slova: uav, autonomní, bezpilotní, dron, manipulátor, stabilizace, STM32, IMU

Překlad názvu: Manipulátor pro bezpilotní autonomní létající dron

Contents

1 Introduction	1	2.2.4 Main processor and IMU	26
1.1 Motivation and problem statement	2	2.2.5 Additional features	27
1.2 Previous work	4	2.2.6 Finished PCB	28
1.3 Related work	4	2.2.7 Testing	30
2 Hardware design	5	3 Software design	33
2.1 Mechanical design	5	3.1 Getting Roll and Pitch angles . .	34
2.1.1 Actuators	6	3.1.1 Data filtration	35
2.1.2 Materials and manufacturing options	7	3.1.2 Angle estimation	37
2.1.3 Manipulator size and work region	9	3.2 Kinematics	38
2.1.4 Digital design in CAD	10	3.2.1 Denavit-Hartenberg parameters	39
2.1.5 Physical prototype	13	3.2.2 Direct kinematics task	41
2.2 Electrical design	14	3.2.3 Inverse kinematics task	41
2.2.1 Power management	15	3.2.4 Choosing which IKT solution will be used	44
2.2.2 USB connection	23	3.3 Servo communication	47
2.2.3 Manipulator and servo connection	24	3.4 ROS Communication	48
		3.4.1 Manipulator ROS node	48
		3.4.2 LLCP ROS node	50

3.4.3 LLCP library	51
3.4.4 Manipulator control program	51
3.5 Full movement pipeline	52
4 Implementation and testing	53
4.0.1 In-flight testing	55
5 Conclusion	57
5.1 Future work	58
Bibliography	59
A Attached CD contents	63

Figures

1.1 Example of a commercial UAV quadcopter - DJI Mavic Pro.	1	2.9 Manipulator prototype in two views.	13
1.2 Example of an MAV quadcopter equipped for science purposes - Holybro X500 used by MRS group at CTU.	2	2.10 Block diagram of the main board controlling the manipulator.	15
1.3 End effector positioning error caused by wind disturbances.	3	2.11 3.3 V LDO (MIC5239) schematic	16
2.1 Schematic representation of an RRR manipulator.	6	2.12 Thermal connection of MIC5239 to the PCB's ground plane	17
2.2 Original mounting hardware for Dynamixel AX-12A servos	7	2.13 Two 12V power supplies tethered in parallel to achieve greater output current.	18
2.3 Deflection simulation of two manipulator links.	8	2.14 Two tethered 12V power supplies based on AP64352Q with surrounding components.	19
2.4 Final dimensions of the manipulator described in this thesis.	9	2.15 High current and frequency loops highlighted in AP64352Q circuit.	21
2.5 Full work region that the end effector can access. [left = side view, right = top view]	10	2.16 High current and frequency loops shown in figure 2.15 highlighted on the final PCB. The top picture represents a copper layout and the bottom one represents a populated PCB model for better illustration.	22
2.6 Exploded view of the finished manipulator design.	11	2.17 Thermal connections for both AP64352Q ICs	22
2.7 Manipulator design iterations. Oldest (left) to current (right).	12	2.18 Schematic for the USB connector	23
2.8 Final manipulator design in multiple views.	12	2.19 Schematic for the servo connector	24
		2.20 Manipulator connector on the PCB	25

2.21 STM32F411RE schematic symbol with named connections to all peripherals mentioned in this thesis. 26	3.6 Manipulator drawing with highlighted axes in each joint 40
2.22 Finished PCB with colored peripherals. 27	3.7 All four possible IKT solutions for the manipulator made for this thesis 42
2.23 Cable tie strain relief for the USB cable. 28	3.8 Topview diagram explaining computations shown in equations 3.8 43
2.24 Finished PCB. Unpopulated (up) and populated (down) version. 29	3.9 2D view of Θ_1 and Θ_2 together with auxiliary paramters 43
2.25 Finished FR4 material links for the manipulator. Both on one panel to save manufacturing costs. 29	3.10 Visual representation of UAV's body collision detection. 45
2.26 Small heatsink attached to the LDO to keep it cooler. 31	3.11 Visual representation of UAV's legs collision detection in XY plane. 46
3.1 Simplified diagram of the main program loop. 34	3.12 ROS communication diagram . 48
3.2 Roll and Pitch angles and IMU axes position relative to the manipulator. Side, front and isometric view. 35	3.13 Manipulator set in the parked position. 49
3.3 Raw and filtered X-axis accelerometer data 36	3.14 High-level movement pipeline diagram for better understanding of the software chapter as a whole. . . 52
3.4 Structure of used 2nd order Butterworth IIR filters 36	4.1 Front-to-end tilting illustration. [left = level, center = tilted back, right = tilted front] 54
3.5 Stabilization algorithm process diagram 39	4.2 Left-to-right tilting illustration. [left = level, center = tilted left, right = tilted right] 54
	4.3 Front view of the manipulator attached to Tarot T650 Sport UAV 55

4.4 Side view of the manipulator attached to Tarot T650 Sport UAV	55
4.5 Detailed view of the manipulator attached to Tarot T650 Sport UAV	56
4.6 Flying UAV with the attached manipulator [left = camera mounted on UAV, right = view from ground]	56

Chapter 1

Introduction

For the last several years there is a significant growth of popularity of Unmanned Aerial Vehicles (UAV). Generally, a UAV is an aircraft of any kind that is flown either autonomously or remotely by a pilot. In both cases, there is no pilot onboard controlling the vehicle.

Such UAVs can be produced in many shapes but the most popular is a Quadcopter, which can be seen in Figure 1.1. Quadcopters consist of a rigid frame in the shape of a letter 'x' or '+' sign with four propellers on its ends. If two of the propellers rotate clockwise and the other two counterclockwise the quadcopter is able to navigate through 3D space in all 6 degrees of freedom¹ (DOF) only by changing the speeds of individual motors. The absence of other moving parts makes quadcopters easy to build, reliable, cheap, and easy to control², hence popular.



Figure 1.1: Example of a commercial UAV quadcopter - DJI Mavic Pro.

¹Only 4 of 6 DOF can be controlled by the quadrotor.

²The ease of control is subject to automatic stabilization of the quadcopter. Without automatic stabilization, a quadcopter would be uncontrollable for humans.

Due to this fact quadcopters are also widely used in the do-it-yourself (DIY) and science communities where they allow fast prototyping and the ability to carry relatively heavy equipment while still maintaining robustness and ease of control at a reasonable cost. These scientific UAVs are often called MAV (Micro Aerial Vehicle) because of their size (Figure 1.2) or Drones because of the sound they produce ³.



Figure 1.2: Example of an MAV quadcopter equipped for science purposes - Holybro X500 used by MRS group at CTU.

1.1 Motivation and problem statement

The thesis is developed under the Multi-robot Systems (MRS) group⁴ at ČVUT. The group is mainly focused on research in the area of aerial, ground, and swarm robotics and cooperates to reach a goal of deployment and application of multi-robot systems in real-world demanding environments. The thesis is mainly motivated by an international robotic competition MBZIRC 2023 held in Abu Dhabi ⁵ where the MRS group is participating. Upon many others, there will be a task where an MAV has to manipulate objects floating

³In English, the word "Drone" originally means "The male of bee". Due to the fact that bees and MAVs make similar sounds when flying, the word, "Drone" became used to describe such MAVs. Source:<https://aeromoments.com>

⁴More information about MRS group can be found under the following link: <http://mrs.felk.cvut.cz>

⁵More information about MBZIRC competition can be found under the following link: <https://www.mbzirc.com>

on water. Since drones used by the MRS group cannot operate touching water, it is ideal to use a lightweight robotic manipulator attached to the bottom side of the drone. The manipulator can then interact with the objects while the drone is hovering a few centimeters above the water level.

This by itself would seem like a complete solution to the problem. But for the manipulator, to work correctly, the drone has to hold a certain position relative to the floating object. Drones equipped with a GPS (Global Positioning System) receiver and other positioning systems like IMUs⁶ (Inertial Measurement Units) or image-based positioning sensors can stay in one position but with a need to tilt the drone in two degrees of freedom. This tilting happens when the drone holds a static position in X and Y coordinates (in the same plane as flight direction) and reacts to outer disturbances like wind. Such behavior can be seen in Figure 1.3 below.

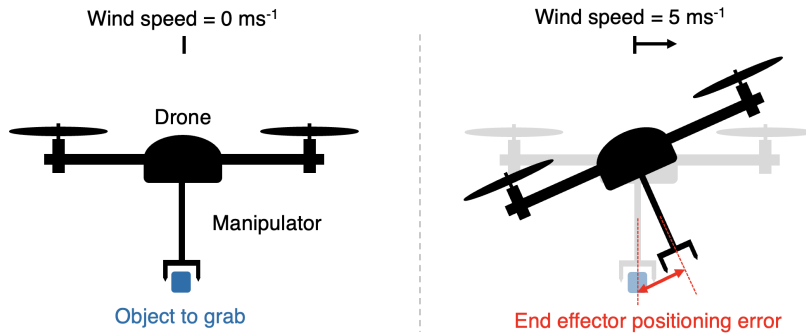


Figure 1.3: End effector positioning error caused by wind disturbances.

To make the end effector hold its position independent of the drone's tilt the manipulator needs to be stabilized (with regards to the tilting of the drone) - which is the goal of this thesis.

The thesis describes the process behind designing and building a lightweight stabilized manipulator for an MAV. In its first part, the thesis focuses on mechanical construction of the manipulator itself and the design of custom electronics together with a PCB (Printed Circuit Board). The second part is focused on software and it includes a control algorithm that collects data

⁶An IMU is present in every drone but since the drone cannot control all of its 6 DOF it cannot keep itself level.

from an onboard IMU and stabilizes the end effector, implementation on STM32F411RE microcontroller⁷, and communication with the drone's onboard computer via ROS (Robot Operating System). Finally, the whole solution is tested airborne in a real world experiment.

1.2 Previous work

This thesis builds on previous work done in the Multi-robot systems group (MRS group) at CTU by David Štych [1]. David designed the first prototype of a UAV manipulator and laid a foundation for this work. This work will be extended by an end effector stabilization functionality based on an onboard IMU.

1.3 Related work

There are many papers dealing with coupled dynamics⁸ of aerial manipulators such as [2] or [3]. The goal in both of these papers is to stabilize a UAV disturbed by the movement of its manipulator. It would be possible to use the mathematical model and data from an IMU to also stabilize the end effector. But since the goal of this thesis is to compute everything outside the main UAV controller this approach is not viable.

Another related field is camera stabilization gimbals based on IMUs and small microcontrollers such as the one described in [4]. The paper describes the usage of IMU data to develop an AHRS⁹ system that is then able to stabilize a small camera gimbal to point in a set direction independent of the aircraft's attitude. A solution like this can be used to develop the manipulator described in this thesis.

⁷Everything needed to stabilize the end effector has to be computed onboard the STM32F411RE microcontroller. This includes collecting data from IMU, filtering the data, computing manipulator kinematics and sending commands to the manipulator's actuators.

⁸Dynamical behavior of two or more coupled objects. In this case, a UAV coupled with a robot manipulator.

⁹Attitude Heading Reference System. A system that provides an aircraft with attitude data based around an IMU

Chapter 2

Hardware design

2.1 Mechanical design

The goal of the mechanical part of this thesis is to design a manipulator capable of navigating its end effector in 3D space. The manipulator has to be lightweight so it can be carried by a small UAV, powerful enough to be able to lift light objects, and robust to eliminate resonance and wobbling caused by the UAV flying.

To satisfy all of the goals stated above a 3 DOF RRR¹ configuration of the manipulator was chosen. 3 (or more) DOF are a mandatory choice if the manipulator has to have a controllable position² in 3D space. Revolute joints are easy to source (the majority of lightweight actuators such as servos or brushless motors are revolute) and usually of lighter weight than prismatic³. Schematic of such RRR manipulator can be seen in figure 2.1.

¹RRR (Revolute Revolute Revolute) manipulator is a three-joint robot that uses revolute joints for its movement.

²General point in 3D space has 6 DOF: 3 for position and 3 for orientation. To control the point's position (3 out of the 6 DOF) three independent actuators are needed.

³Prismatic joints are moving on a linear trajectory. Prismatic joints are often built from revolute actuators with added gearboxes and other mechanics which make the prismatic actuator heavier than a revolute one.

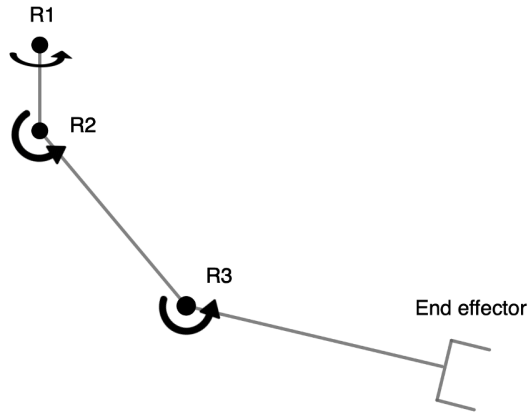


Figure 2.1: Schematic representation of an RRR manipulator

2.1.1 Actuators

As actuators for the revolute joints mentioned above Dynamixel AX-12A servos were chosen. The main reason is that they are widely used across MRS group platforms, are easy to source, and can satisfy all of the requirements for a lightweight powerful joint. The most important specifications of the servos are written in table 2.1. The servos also feature original plastic mounting brackets (figure 2.2) which can simplify the mechanical design of the manipulator.

AX-12A specification	value
Weight	53.5 g
Resolution	0.29°
Range of motion	0 to 300°
Stall torque	1.5 Nm at 12 V
No load speed	59 rpm
Input Voltage	9 to 12 V

Table 2.1: Dynamixel AX-12A servo specifications [11]

■ 2.1.2 Materials and manufacturing options

Materials and fabrication techniques were chosen with respect to the ability to construct the manipulator at home or in a lab with minimal help from other companies and with the possibility to iterate designs and service the finished manipulator in the shortest time possible.

Firstly it is convenient to use the Dynamixel AX-12A's original mounting hardware since it can be easily mounted to the actuators with M2.5 bolts. The brackets are made of injection molded ABS plastic which makes them strong, lightweight, and resistant to temperatures of up to $70\text{ }^{\circ}\text{C}$ before softening⁴.

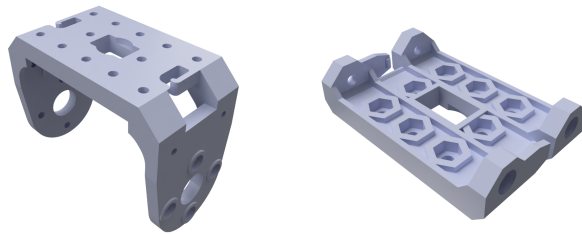


Figure 2.2: Original mounting hardware for Dynamixel AX-12A servos

The most used manufacturing technique in this thesis is 3D printing from PET⁵ material. 3D printing is the ideal technology to create prototypes of sufficient strength and quality in a home or lab environment. It also allows for quick manufacturing of replacement parts.

Problematic are flat and relatively long parts used for the manipulator's links which are prone to bending when 3D printed (see figure 2.3). For these parts, an FR4 material was used. FR4 is a glass fabric and epoxy resin composite plated with copper from both sides used for PCB fabrication. Its main advantages are a great stiffness-to-weight ratio and the possibility to print electrical circuits directly onto the material (later used in section

⁴Information about ABS material was gathered from <https://www.matweb.com/search/DataSheet.aspx?MatGUID=3a8afcdac864d4b8f58d40570d2e5aa&ckck=1>

⁵PET (polyethylene terephthalate) is a thermoplastic widely used in the hobbyist 3D printing field mostly for its good mechanical properties and ease of use.

2.1.4). To compare the 3D printed PET manipulator link with that of an FR4 material a simple simulation in Autodesk Fusion 360 software was conducted (figure 2.3).

In the simulation, one link is made of FR4 material and the other one from solid⁶ PET material. Both are fixed on both ends and loaded by a force of 200 N in the middle of the 100 mm part. Observed is the bending of both parts. It is clearly visible (figure 2.3) that the FR4 part is about 2.5 times stronger than that of the PET material.

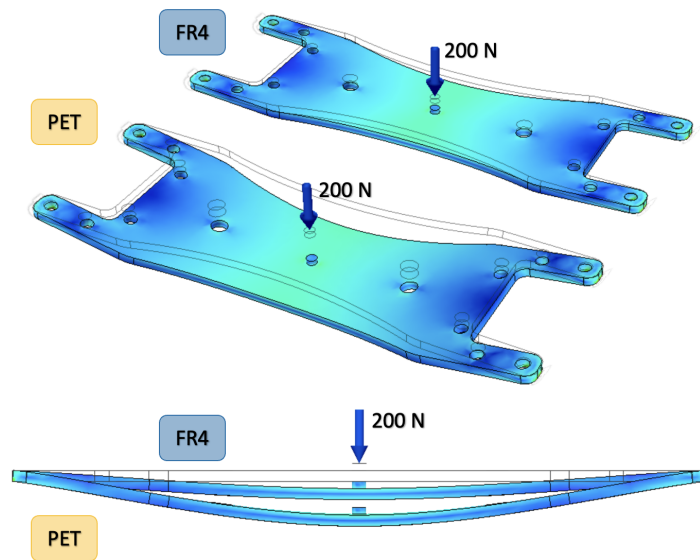


Figure 2.3: Deflection simulation of two manipulator links.

In the previous work, [1] CNC milled⁷ carbon fiber plates were used. Despite carbon fiber's better weight and strength properties, FR4 was chosen for its ease of use and the advantage of conducting electrical signals and power.

⁶In case of 3D printing even if the material is printed with 100% infill (being solid) it does not reach the same mechanical properties as 100% filled injection molded part of the same material.

⁷CNC (Computer Numerical Controlled) milling is a machining process utilizing computer controlled mill to subtractively fabricate products out of stock material.

2.1.3 Manipulator size and work region

Sizewise the manipulator is heavily influenced by the previous work [1] where almost ideal dimensions were proposed for an MRS group application. These parameters were slightly changed to accommodate for differently oriented servos. The final dimensions of the manipulator can be seen in figure 2.4 below.

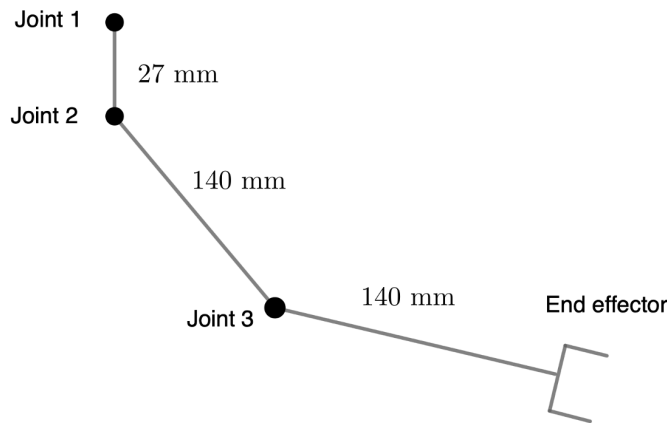


Figure 2.4: Final dimensions of the manipulator described in this thesis.

After considering the size of the manipulator (figure 2.4) and travel limits of all the joints, the accessible work region is a reduced intercept of two spheres with a 277 and 150 mm diameter centered in the first joint (figure 2.5). In chapter 3 this space is further reduced by software to avoid collisions with the drone's body.

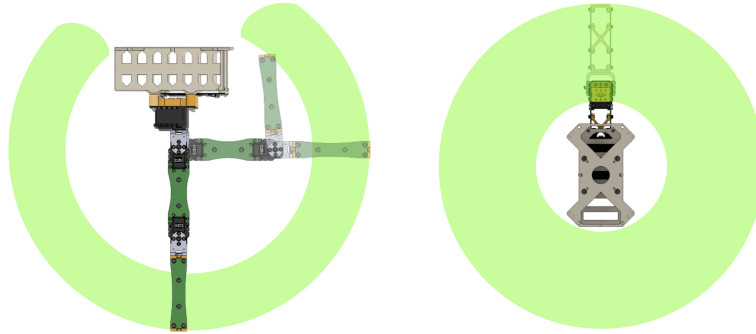


Figure 2.5: Full work region that the end effector can access. [left = side view, right = top view]

■ 2.1.4 Digital design in CAD

The whole manipulator mechanics were designed and modeled in Autodesk Fusion 360 CAD⁸ software. Such software allows for quick prototyping, testing and simulations on a computer before the real product is manufactured. This way both costs and time are saved while designing a product.

For this thesis, CAD was used mainly for designing 3D printed parts, preparing PCB outlines, testing range of motion, and determining measurements for kinematic equations in section 3.2. It was also used to create all of the 3D drawings presented in this thesis.

■ The design process

The manipulator was designed in several steps which will be described in the following paragraphs. For reference, figure 2.6 shows all of the further mentioned manipulator parts in an exploded view. Yellow parts together with the battery holder are 3D printed, and black and green flat parts are made out of FR4 material (see section 2.1.2).

⁸CAD (Computer Aided Design) is a software which allows modeling of real 3D objects using a computer.

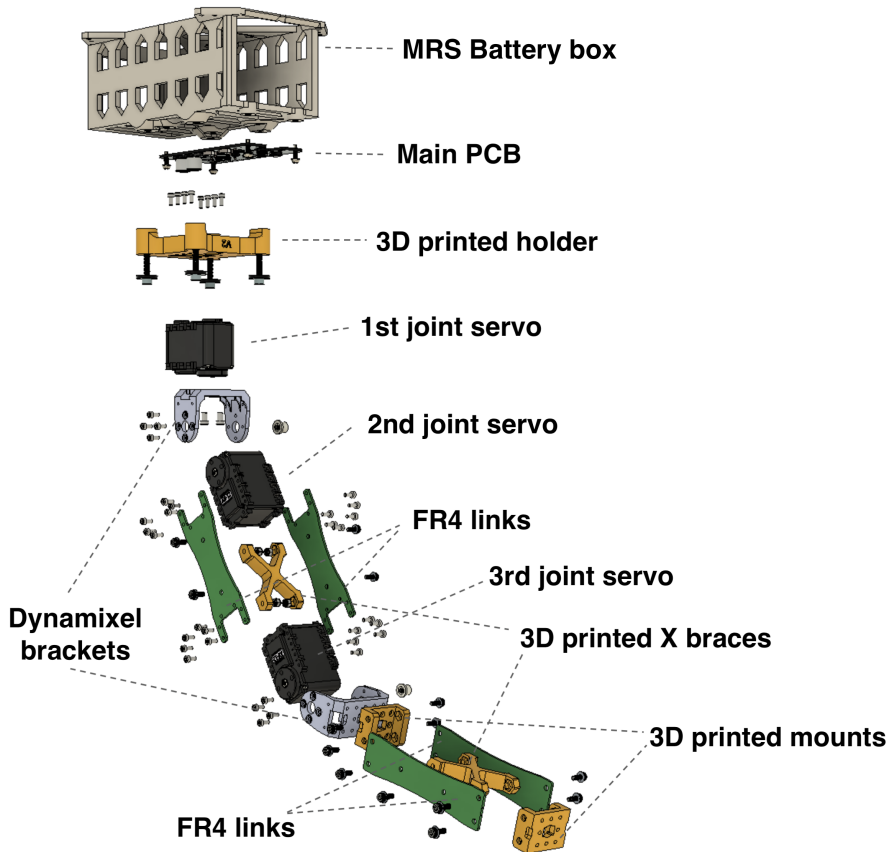


Figure 2.6: Exploded view of the finished manipulator design.

At first, the main PCB outline and the 3D printed manipulator holder were designed as it makes for the only interface between the manipulator and the MRS battery box⁹ used on the target UAV. Then all of the servos were placed in their final locations together with the dynamixel brackets. After that, all of the manipulator links and their 3D-printed mounts were designed to hold the manipulator together. To fabricate the links, as mentioned in section 2.1.2, an FR4 material was used. This allowed printing circuits directly on the links, which results in clean cable management and the possibility to get electrical signals cleanly to the end effector.

At this point, thanks to the long and thin PCB links, the manipulator was prone to breaking so the 3D-printed bracing was added. In the last

⁹A 3D printed battery box designed by the MRS group at CTU.

step, the MRS battery box was redesigned to support easy mounting of the manipulator with several bolts and locknuts.

The final design (figures 2.6 and 2.8) went through many iterations, some of which can be seen in figure 2.7



Figure 2.7: Manipulator design iterations. Oldest (left) to current (right).

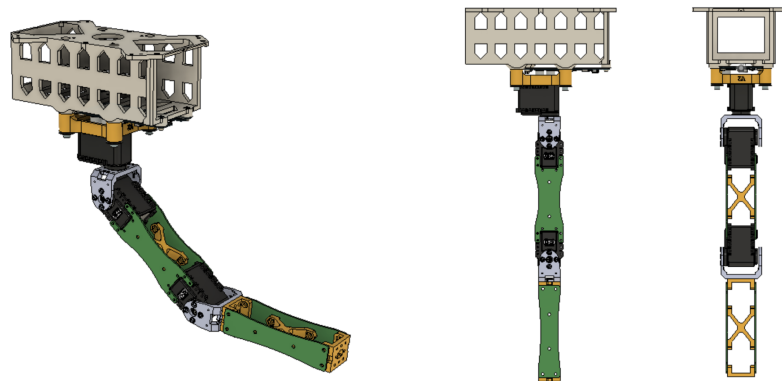


Figure 2.8: Final manipulator design in multiple views.

2.1.5 Physical prototype

The assembled prototype can be seen in figure 2.9. After assembling the manipulator came out lightweight and robust as stated in the goals of this thesis. What was not mentioned in the design part but is needed for assembly is the usage of self-locking nuts¹⁰ or a thread-locking glue to ensure the manipulator stays intact when attached to a vibrating UAV. Also on the left side of the figure, a PCB is shown. Its design will be described in chapter 2.2 of this thesis.

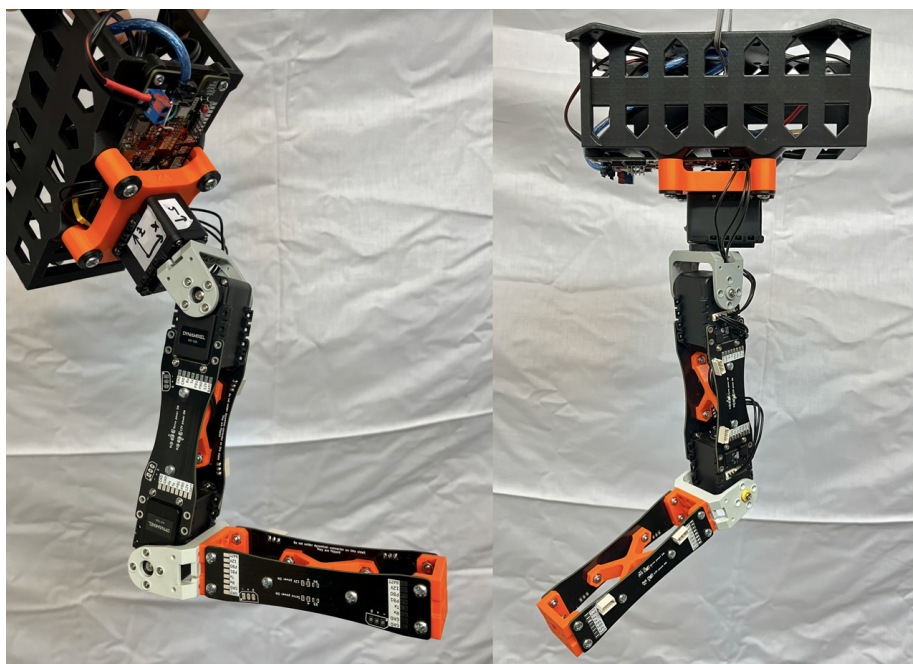


Figure 2.9: Manipulator prototype in two views.

¹⁰Self-locking nut is a special nut with a silicone ring that is not prone to loosening by vibrations.

2.2 Electrical design

The goal of this part of the thesis is to show the process behind designing a PCB (Printed Circuit Board) and all of its components needed to run the manipulator shown in section 2.1. Firstly the design requirements are stated, next a block diagram is created, to support all of the requirements, and lastly, the design of each block from the diagram is described. Additionally for parts of the block diagram where thermal or layout considerations are needed it is shown how they were met.

Design requirements

- The main processor used has to be an STM32F411RE.
- The PCB will be powered from the UAV's battery and has to support voltages in the range of 14 – 25.2 V.
- All of the measurement and processing has to be done onboard (an IMU needs to be on the board).
- The board has to have several MOSFET switched outputs and GPIO¹¹ pins are free to support future mounted peripherals.
- A USB (Universal Serial Bus) connection has to be present for communication with the main UAV computer.

Block diagram

To support all of the mentioned requirements a block diagram is created (figure 2.10). All power rails are shown as colored arrows and data buses are shown as white arrows. According to the diagram (2.10), the board can be powered from the UAV's battery, can be connected to the drone's computer via USB, can be programmed through an SWD¹² interface, measures and processes all of the needed data onboard and provides the needed GPIO and MOSFET switched pins.

¹¹GPIO stands for General Purpose Input Output and is usually connected with general purpose microcontroller pins.

¹²SWD is a Serial Wire Debug interface used by STLink programmers to program STM32 processors.

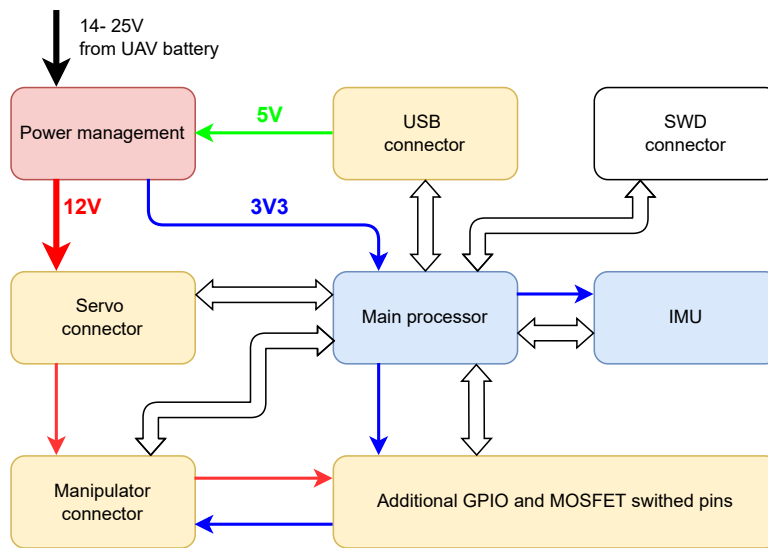


Figure 2.10: Block diagram of the main board controlling the manipulator.

■ 2.2.1 Power management

As shown in the block diagram (figure 2.10) the power management needs to convert incoming battery voltage (14 to 25.2 V) to 12 V for servos, manipulator accessories and GPIO, together with 3.3 V needed for all other elements of the board. Another feature of the power management is the capability of powering the 3.3 V rail from a provided USB voltage (5 V) if the main power from the UAV's battery is disconnected.

3.3 V rail and USB power input

To provide 3.3 V from the UAV's battery an LDO¹³ MIC5239 regulator is used. According to its datasheet [13] the IC can supply continuous 500 mA current with input voltages ranging from 2.3 to 30 V, which makes it ideal for this application. The regulator is connected to the board based on an example connection shown in the datasheet [13].

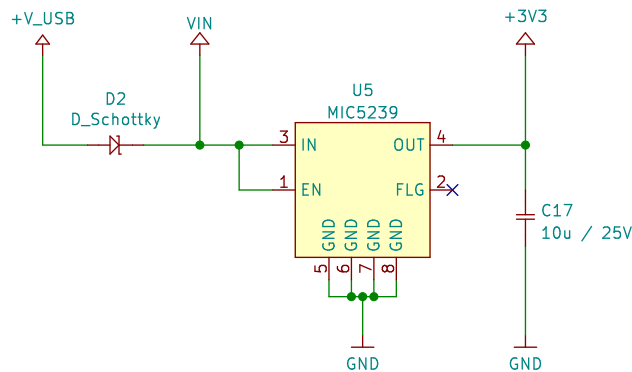


Figure 2.11: 3.3 V LDO (MIC5239) schematic

The USB voltage mentioned above (2.2.1) is connected through a *DIOTEK SEMICONDUCTOR SK110* Schottky diode (D2) with a typical forward voltage $V_F = 0.6$ V [15]. This allows for connecting both the input voltage V_{IN} and the USB voltage $+V_{USB}$ at the same time and having them switch automatically to the bigger one¹⁴. The only problem that could occur with this solution is a low voltage on the MIC5239 input due to the diode's V_F and low voltage on the USB bus ($+V_{USB}$).

The $+V_{USB}$ voltage can vary between 4.75 and 5.25 V [14] and the diode's forward voltage can be in the range of 0.5 to 0.85 V [15]. Then in the worst case, the MIC5239 would get an input voltage of $4.75 - 0.85 = 3.9$ V. The LDO's dropout voltage is set to 350 mV [13] hence even in the worst case

¹³LDO (Low DropOut) voltage regulator is a linear regulator with a low voltage drop from its input to its output - typically between 0.5 and 1.5 V. For instance, a 3.3 V LDO regulator with a 0.5 V dropout voltage can be used with inputs as low as 3.8 V.

¹⁴When both V_{IN} and $+V_{USB}$ are present, the power for MIC5239 is drawn from V_{IN} . If the input voltage V_{IN} is disconnected, the diode opens and power for the MIC5239 is sourced from $+V_{USB}$

the regulator would remain operational and stably deliver its output voltage of 3.3 V.

The last part is the thermal design. At full load (0.1 A) and maximum allowed battery voltage (25.2 V), the LDO will need to dissipate 2.19 W of thermal power¹⁵. The junction to ambient thermal resistance for MIC5239 is $\Theta_{JA} = 80 \text{ }^\circ\text{C/W}$ [13] which is unacceptable (the IC would overcome its maximum junction temperature $T_{J,max} = 125 \text{ }^\circ\text{C}$). This problem is solved by connecting the IC's GND pins (which can be used to sink heat from the IC [13]) to the PCB's ground plane through multiple vias (see figure 2.12). The heat is then sunk to the ground plane and dissipated throughout the whole PCB ground area.

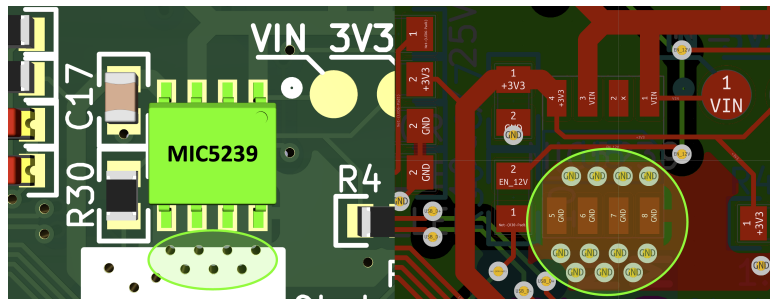


Figure 2.12: Thermal connection of MIC5239 to the PCB's ground plane

■ 12V rail

The 12 V rail has to deliver a stable 12 V output with a peak current draw of 5.5 A and stable current draw of 4 A¹⁶. Originally the 12 V rail was designed around AOZ1269 which is capable of delivering a continuous output of 12 A [16], but after manufacturing the PCB and populating it with parts, the IC never worked properly. After a consultation with the thesis supervisor and

¹⁵The dissipated thermal power for linear regulators is computed as $V_{drop} \cdot I$, where V_{drop} is the voltage drop across the regulator and I is the current supplied by the regulator. In this concrete case, the dissipated power is equal to $(25.2 \text{ V} - 3.3 \text{ V}) \cdot 0.1 \text{ A} = 2.19 \text{ W}$.

¹⁶The peak current is given by the servo's power requirements. If a servo is stalled the stall current is 1.5 A [11]. If all three of the servos are stalled the total current consumption is 4.5 A. The remaining 1 A to the stated 5.5 A is reserved for powering other peripherals present on the PCB. In a normal (non-stalled) state each servo should remain under 900 mA current consumption [11].

numerous failed attempts to make the AOZ1269 functional it was decided to use an AP64352Q instead.

AP64352Q is a simple synchronous step-down converter widely used and tested in the MRS group. Its voltage input range is between 3.8 and 40 V and requirements for external parts and layout complexity are low (compared to the previously used AOZ1269) [17]. The only problem is that the regulator is not able to supply more output current than 3.5 A continuously, which does not satisfy the previously stated requirements for the manipulator. The solution to this problem is tethering two of these regulators in parallel.

Since both of the converters are not manufactured with perfectly the same parameters one will have a slightly higher output voltage than the other. This will cause that all of the output power will be provided by the converter with the higher output voltage. To solve this problem, two small value resistors are used to connect the converters (see figure 2.13) so they can share the load current [18].

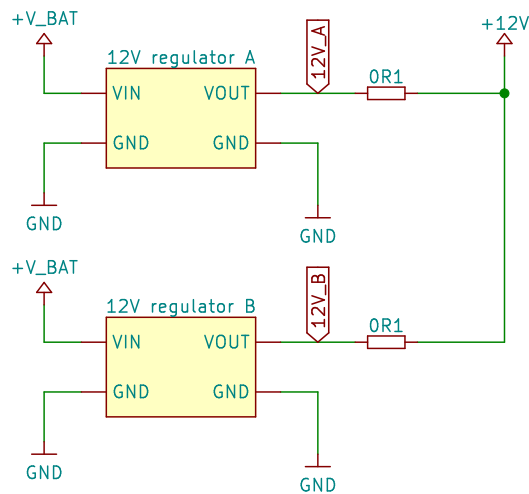


Figure 2.13: Two 12V power supplies tethered in parallel to achieve greater output current.

This connection builds on the Ohm's law. When the output current of one converter starts to rise a significant voltage drop occurs on its tethering resistors (labeled "0R1" in figure 2.13) and lowers the converter's output voltage

contribution to the tethered 12V line. This will allow the lower voltage power supply to contribute to the main power output too. This solution is valid for a low-cost parallel DC to DC converter connection with low requirements for uniform power distribution between the two power supplies [18], which is exactly the situation in this thesis¹⁷. The final schematic with two AP64352Q regulators connected in parallel can be seen in figure 2.14 below.

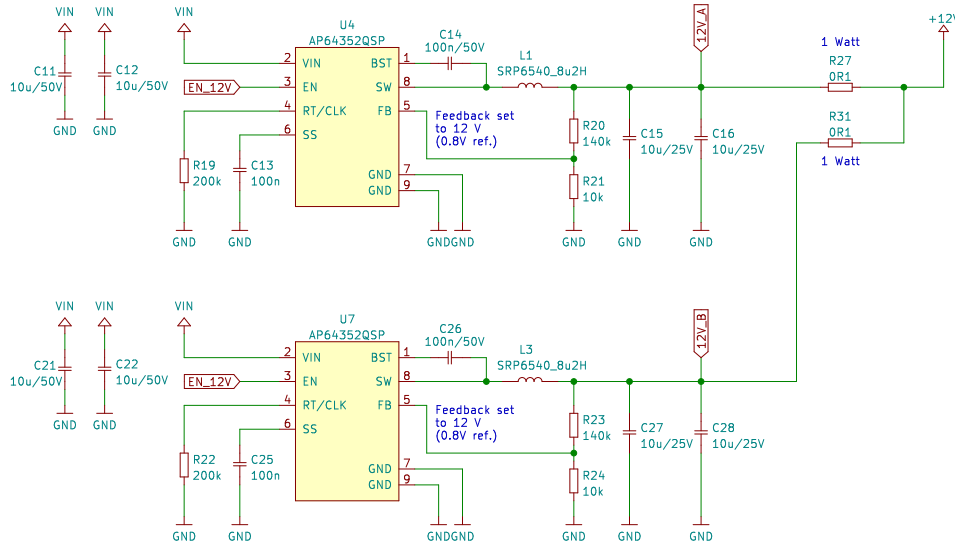


Figure 2.14: Two tethered 12V power supplies based on AP64352Q with surrounding components.

The schematic follows recommendations from the IC’s datasheet [17] with a different inductor ($8.2 \mu H$ SRP6540 used by the MRS group, which results in smaller output voltage ripple) and smaller output capacitors ($2 \times 10 \mu H$ instead of $2 \times 22 \mu H$). The lack of output capacitance is later substituted with $2 \times 100 \mu F$ capacitors on the tethered 12V line (common for both regulators).

Feedback resistors $R20$ and $R21$ (and $R22$ with $R23$ respectively) are calculated from knowing the IC’s feedback voltage $V_{fb} = 0.8 V$ [17] and a resistor voltage divider formula as follows:

¹⁷If the maximum peak current draw is $5.5 A$ and two $3.5 A$ converters are connected in parallel the worst case load balance can be set to 64/36%. In this case one of the regulators will deliver its 100% output and the other will be loaded with 57% of its maximum output ($1 \cdot 3.5 A + 0.57 \cdot 3.5 A = 5.5 A$). This is far from a uniform distribution (50/50%) and hence perfectly achievable by using two resistors.

$$V_{fb} = V_{out} \cdot \frac{R21}{R20 + R21} \rightarrow R21 = \frac{V_{fb} \cdot R20}{V_{out} - V_{fb}} \quad (2.1)$$

$R20$ is then set to a $140\text{ k}\Omega$ value and $R21$ is computed based on the formula above.

$$R21 = \frac{0.8\text{ V} \cdot 140\text{ k}\Omega}{12\text{ V} - 0.8\text{ V}} = 10\text{ k}\Omega \quad (2.2)$$

The last problem to consider is a varying output voltage that is dependent on the current supplied to the load. The higher the current the higher the voltage drop on the tethering resistors, hence lower voltage on the main 12 V rail. In the worst case the voltage drop V_{drop_max} will equal to

$$V_{drop_max} = I_{max} \cdot R_{tether} = 3.5\text{ A} \cdot 0.1\text{ V} = 0.35\text{ V} \quad (2.3)$$

where I_{max} is the maximum output current per regulator and R_{tether} is the value of the tethering resistors. That results in an output voltage of $V_{out_min} = 11.65\text{ V}$ ($12 - 0.35\text{ V}$). In this concrete application where the main purpose of the 12 V power supply is to power the Dynamixel AX-12A, which can operate at even lower voltages [11] and other peripherals which are not voltage sensitive is this solution is acceptable.

Similarly to the voltage drop a maximum dissipated power for the resistors P_{max} can be calculated to size their cooling and size requirements.

$$p_{max} = R_{tether} \cdot I_{max}^2 = 0.1\text{ V} \cdot 3.5^2\text{ A} = 1.22\text{ W (per resistor)} \quad (2.4)$$

12V rail layout

The layout goal for the 12V power supply is to minimize EMI¹⁸ produced by the switching regulators and to keep the temperature low for both ICs. To reduce EMI it is needed to find all loops where high currents of high frequencies can occur and minimize their area on the PCB. In each IC two current paths could cause potentially high EMI (see figure 2.15).

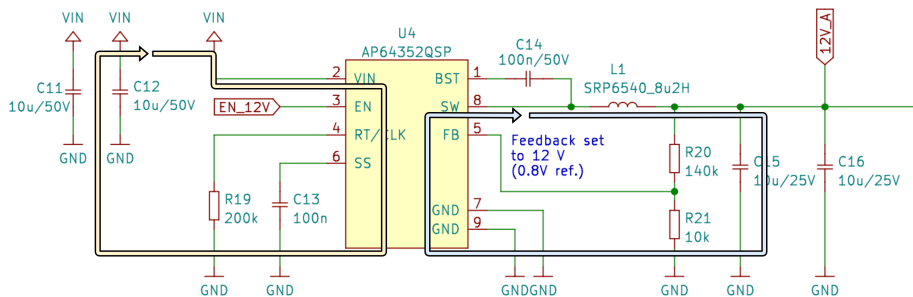


Figure 2.15: High current and frequency loops highlighted in AP64352Q circuit.

The highlighted loops are then minimized by a carefully chosen layout on the PCB. The final solution is shown in figure 2.16 below. It can be seen that all of the loops have minimal area and are backed by a ground plane on the other side of the PCB which furthermore reduces EMI. Also, no EMI-sensitive components or signal traces are routed close to the switching power supplies area.

¹⁸EMI stands for ElectroMagnetic Interference which represents an unwanted electromagnetic noise caused by an external source. In this case by the inductor and large current loops.

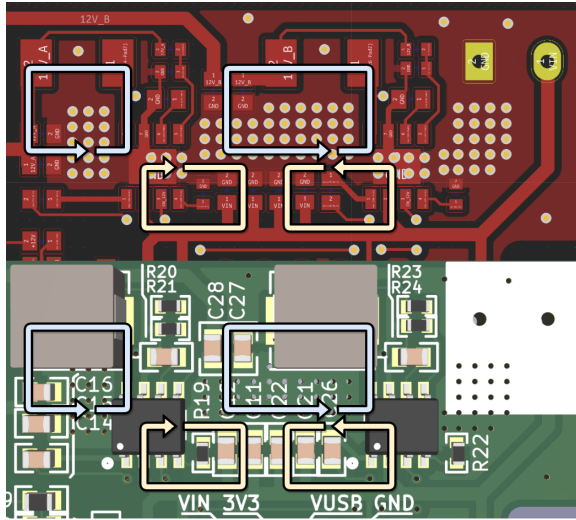


Figure 2.16: High current and frequency loops shown in figure 2.15 highlighted on the final PCB. The top picture represents a copper layout and the bottom one represents a populated PCB model for better illustration.

The thermals are solved similarly to the 3V3 rail (see figure 2.12) with a copper-filled zone on the top side of the PCB connecting both of the IC's exposed pads¹⁹ which is stitched to the bottom ground plane with many vias as shown in figure 2.17.

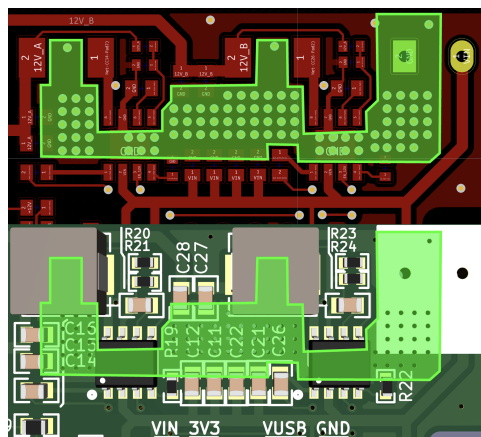


Figure 2.17: Thermal connections for both AP64352Q ICs

¹⁹An exposed pad is a solderable metal surface on the bottom of an IC. Usually, it is used for conducting heat out of the package or for high current applications.

2.2.2 USB connection

The USB connection serves the purpose of communication between the manipulator and the UAV's computer. Since the used processor (STM32F411RE) uses USB 2.0 protocol with Full speed standard [19] the communication speed is 12 *Mbit/s* with a signal rise time of 4 – 20 *ns*.

If the PCB trace between the processor and the USB connector remains under 80 *mm* long²⁰ it is not needed to treat it as a high-speed transmission line hence no need for impedance control or line termination. In the case of this thesis, the USB data trace length is approximately 48 *mm* so the connection to the processor is a regular microstrip pair with overcurrent protection resistors as seen in figure 2.18

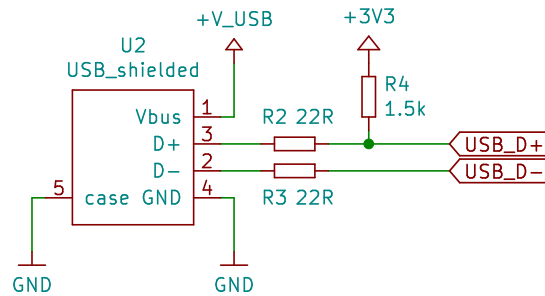


Figure 2.18: Schematic for the USB connector

²⁰Signal speed on an FR4 PCB is around $100 \cdot 10^6$ *m/s* which can be written as 10 *cm/ns*. At 4 *ns* rise time of the USB signals the critical length for high-speed design is 40 *cm* (10 *cm/ns* · 4 *ns*). To be safe 20% of this value is considered a high-speed design threshold.

2.2.3 Manipulator and servo connection

To connect the PCB to the manipulator two connectors are used. The connector for the servos is a generic MOLEX 22-03-5035 used by the Dynamixel AX-12A ecosystem [11] which is connecting 12V power and data to the servos. The power to the servos can also be disabled by software (via AOD4186 MOSFET connected in series with the connector) to enable the user to remotely reset the servos in case of failure. Below is a schematic for the servo connector (figure 2.19). Note the two MOSFETs ($T4$ and $T3$) chained in the schematic. That is because $T3$ (AOD4186) is not able to fully open at 3.3V so it is controlled by a 12V signal which is operated by $T4$ that can be fully opened by 3.3V.

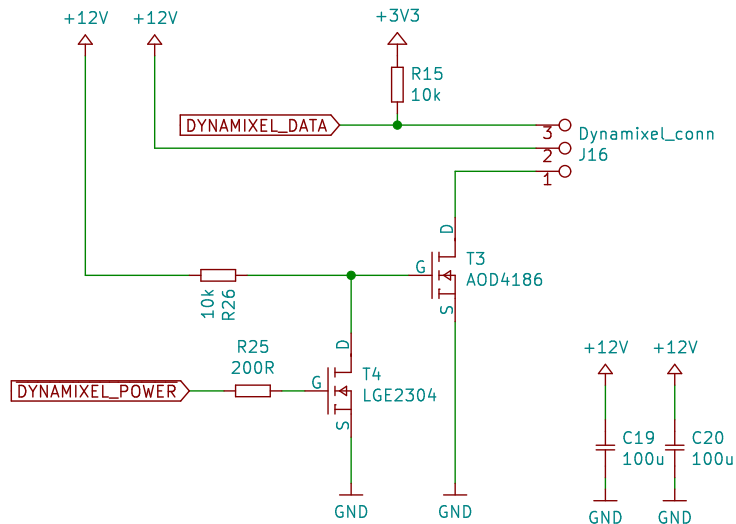


Figure 2.19: Schematic for the servo connector

As described in section 2.1.4 the manipulator is equipped with PCB material links to be able to carry signals and power to the end effector. To connect the links to the PCB a proprietary SMD connector is used. It is an 8-pin connector carrying battery voltage (VIN), 12V power, two ground connections (GND), UART line (Tx and Rx), and two user GPIO pins connected to $PB1$ and $PB0$ on the main processor. On the final board the connector looks as follows:

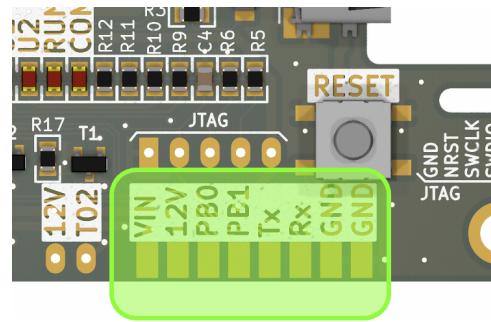


Figure 2.20: Manipulator connector on the PCB

2.2.4 Main processor and IMU

The main processor (STM32F411RE) is a 32-bit ARM chip with 512 kB of RAM manufactured by ST Microelectronics running at 96 MHz. The main advantages of this chip are its low cost, ease of operation, computing speed, and many built-in peripherals like I2C or USB that are used for this thesis. Electrically the processor is connected to 3.3 V power with added decoupling capacitors and then to all of the peripherals according to the schematic shown in figure 2.21.

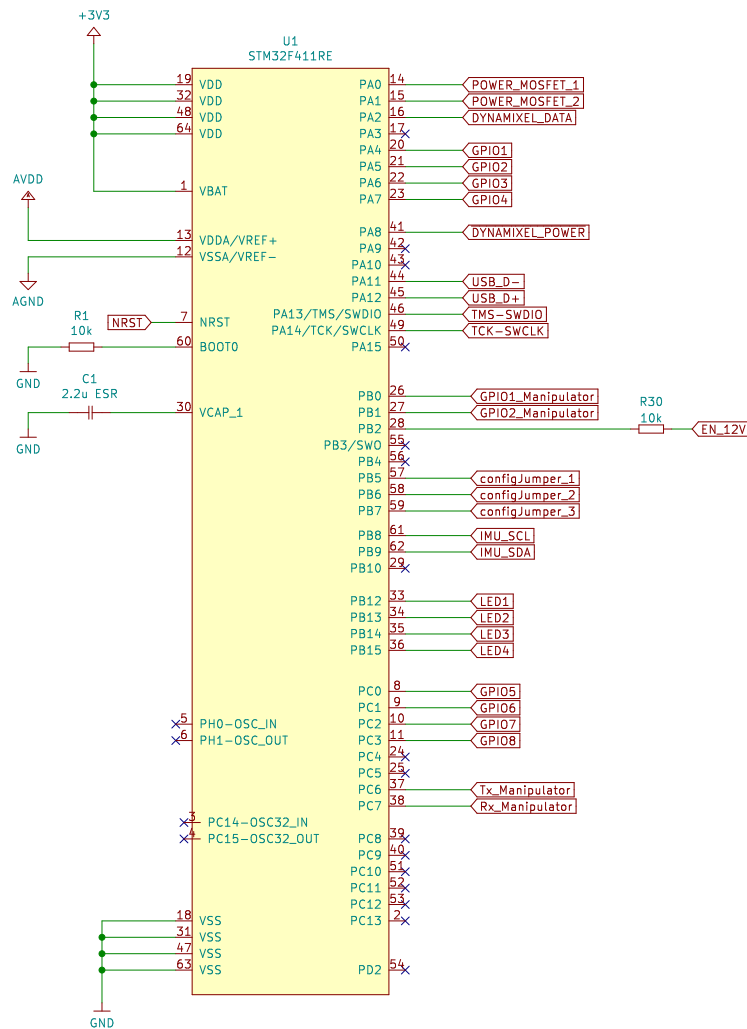


Figure 2.21: STM32F411RE schematic symbol with named connections to all peripherals mentioned in this thesis.

2.2.5 Additional features

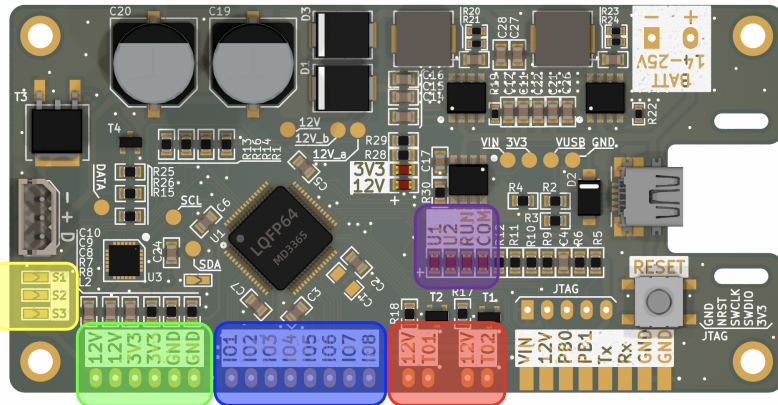


Figure 2.22: Finished PCB with colored peripherals.

Besides all of the components mentioned above the board is equipped with several user features and connections for future expansion. First are 8 additional GPIO pins named IO1 to IO8 (see the blue box in figure 2.22). These are connected directly to the main processor (see figure 2.21) and can be used as a general GPIO, ADC, or timer output. To power additional hardware, all of the present voltages (12 V and 3.3 V) are available as outputs from the board (green box in figure 2.22). This allows for easy connection of other hardware. Electrical parameters of these voltage outputs can be found in section 2.2.1. The board also offers two MOSFET (part number: LGE2304) switched outputs (red box in figure 2.21) which are located near a 12 V power output. This enables the user to connect two MOSFET switched 12 V devices with a maximum current draw of $2 \times 3 \text{ A}$ ²¹.

For user setup, three solder bridges are present on the board (yellow zone in figure 2.22). The user can close the bridge by adding solder to it and easily change the settings or parameters of the program loaded in the main chip without needing to reprogram it. The last feature is indication LEDs (purple

²¹The maximum current draw will be limited by the maximum remaining supply current in the main 12V power supply before saturation.

zone in figure 2.22) used for indicating whether the device is running and communicating with the main UAV computer. Two LEDs are programmable by the user.

Mechanically the board features 4 mounting holes fitting M3 bolts and strain relief for the connected micro USB cable. The strain relief consists of two slots in the PCB with a cable tie (see figure 2.23).

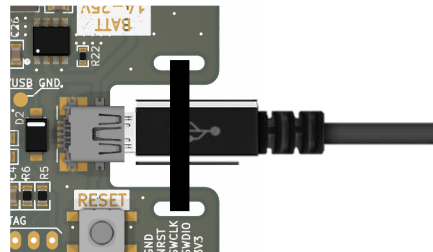


Figure 2.23: Cable tie strain relief for the USB cable.

■ 2.2.6 Finished PCB

The finished PCB can be seen in figure 2.24 followed by a panel containing both types of PCB links mentioned in section 2.1.2.

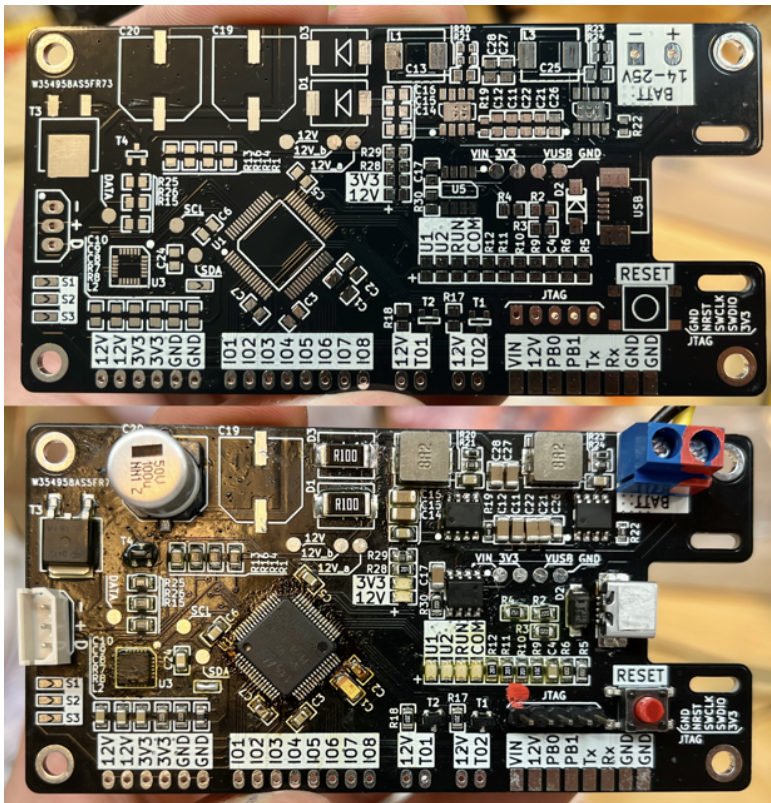


Figure 2.24: Finished PCB. Unpopulated (up) and populated (down) version.

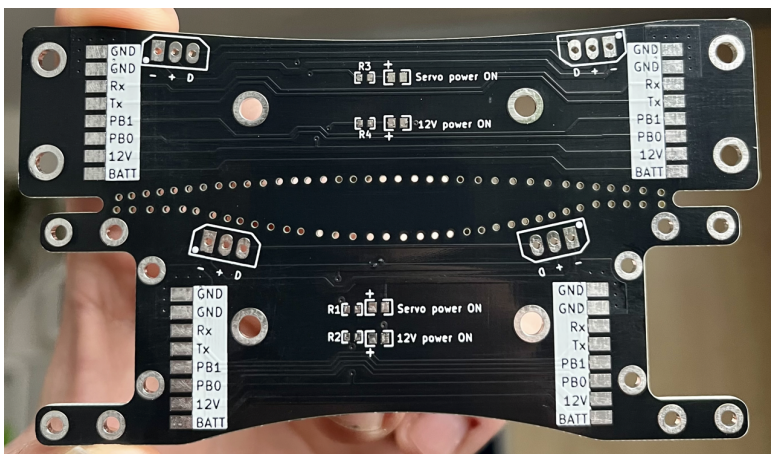


Figure 2.25: Finished FR4 material links for the manipulator. Both on one panel to save manufacturing costs.

■ 2.2.7 Testing

After manufacturing and assembling the board it was tested for all important electrical specifications. All of the tests are listed below:

- Maximum input voltage of 25.2 V. [PASSED]
- Minimum working input voltage of 14 V. [PASSED]
- Stable 12 V rail under 3 A load. [PASSED]
- Stable 3.3 V rail under different input voltages. [PASSED]
- Load balancing between the two 12 V buck converters²² [PASSED]
- Thermal testing [CORRECTED] (see the section below)

■ Thermal testing

The first components that can get hot are the 12V power supplies. For testing the board was loaded with a 12 V, 36 W LED strip which was drawing 3 A from the 12 V rail. The input voltage was set to 15.3 V (the board was drawing around 2.64 A) and the ambient temperature was 22 °C. After 10 minutes with no active cooling (the board was mounted in the manipulator in a room with no airflow) the regulators got to 70 °C²³. For another 20 minutes, the input voltage was raised to 25 V (the current draw was approx. 1.67 A). The final temperature (total testing time was 30 minutes continuous) got to 74 °C, which is well within the specification of the used AP64352Q [17]. The balancing resistors were also measured and stayed within 60 °C.

Another component producing heat is the 3.3 V LDO regulator. This test was running at the same time as the 12 V regulator test to simulate the worst-case scenario. The LDO was loaded by a running processor and all of its peripherals. In the first 10 minutes (input voltage set to 15.3 V) the regulator got to 81 °C. At the end of the test (another 20 minutes with input voltage set to 25 V) the LDO got to a temperature of 113 °C which is almost at the regulator's limit of 125 °C [13]. Considering that in summer

²²The load balancing was not expected to be 50/50 % (see section 2.2.1). When a light load is attached (under 0.5 A) the load balancing is almost 50/50 %. When a higher load is attached (tested with 3 A) the balance becomes about 60/40 % - which lies within specification.

²³All of the mentioned temperatures were measured by a contactless thermometer GM320.

the ambient temperatures could rise to 40°C the regulator was not safe to operate. As a solution, a small self-adhesive aluminium heatsink was added (figure 2.26) to the LDO to keep it cooler. After running it for another 15 minutes with an input voltage of 25 V the temperature hasn't exceeded 80°C hence making the board reliably functioning outside in the summer.

This is why a [CORRECTED] mark was given above. For future iterations, the regulator could be connected behind the 12 V regulator to reduce the wasted heat (described in section 2.2.1) and make the regulator cooler without needing a heatsink.

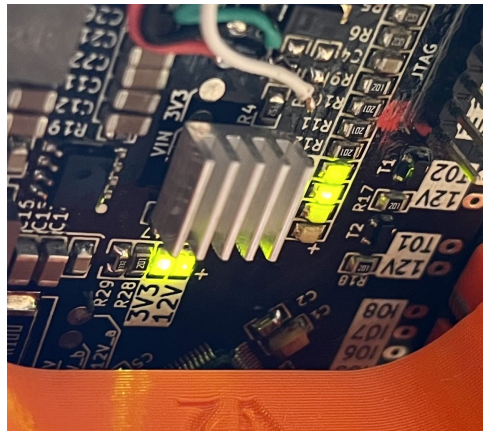


Figure 2.26: Small heatsink attached to the LDO to keep it cooler.

The last component prone to heating is the MOSFET switching power to the servos. As it's switched on resistance $R_{DS(ON)}$ is around $24\text{ m}\Omega$ when operating at 12 V [21] the wasted heat at a 3 A load will be only 0.22 W ²⁴ This fact was tested by loading the MOSFET with an LED strip drawing 3 A . After 10 minutes the MOSFET was just lukewarm to the touch hence safe to operate.

Note that all of these tests were performed under the worst-case scenario with no airflow around the components. In real-world use, while mounted on the aircraft, high current consumption can only occur while flying (when landed the manipulator has to be turned off) where there is a sufficient amount of airflow to cool the components better than in the tests performed.

²⁴Wasted heat power computation: $P = R \cdot I^2 = 24\text{ m}\Omega \cdot 3^2\text{ A} = 0.216\text{ W}$.



Chapter 3

Software design

This chapter is focused on the key thoughts and methods used to develop firmware for the manipulator. It is running on an STM32F411RE on the board described in chapter 2.2. The firmware is responsible for the manipulator's stabilization, collision prevention, and communication with the UAV's computer running ROS (Robot Operating System) via UART¹, and for all other low-level control such as end effector operation, servo communication, status LED indication, etc.

As a result, the UAV is only sending target position commands to the manipulator and receives telemetry. All of the computations needed to move the manipulator are done onboard on the manipulator itself. A simplified diagram of the main program loop can be seen in figure 3.1 of which every part will be described in detail throughout this chapter.

¹UART (Universal Asynchronous Receiver Transceiver) is a simple serial communication protocol used by the majority of devices.

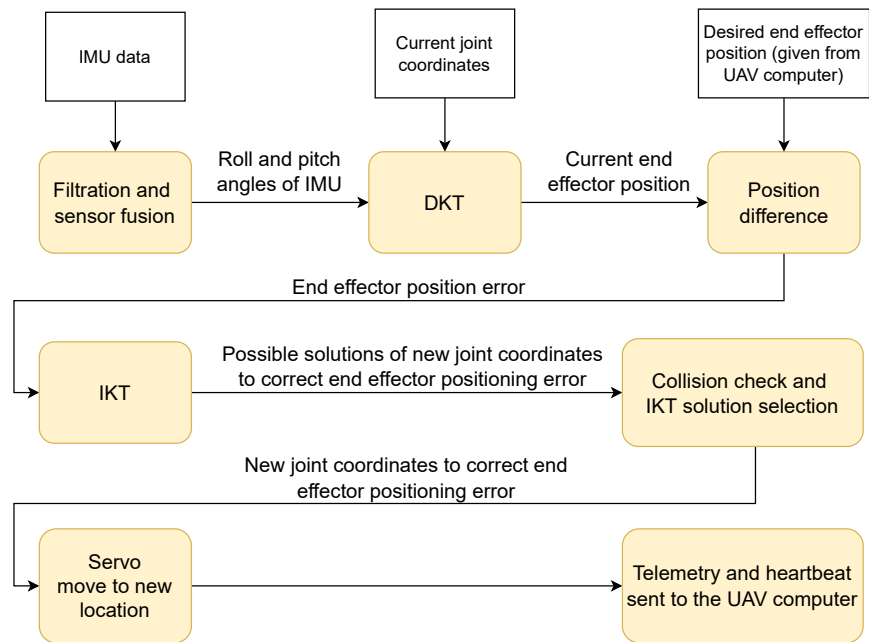


Figure 3.1: Simplified diagram of the main program loop.

3.1 Getting Roll and Pitch angles

As shown in Figure 3.1 it is needed to estimate Roll and Pitch angles of the manipulator based on IMU data. According to the IMU's datasheet [5] it provides acceleration and angular velocity around all three axes. These are shown in Figure 3.2 in their position relative to the manipulator.

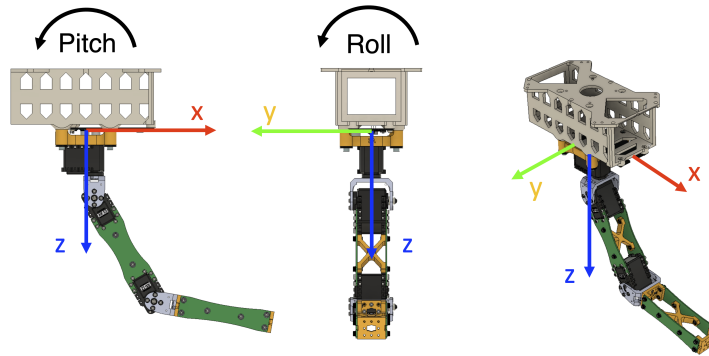


Figure 3.2: Roll and Pitch angles and IMU axes position relative to the manipulator. Side, front and isometric view.

■ 3.1.1 Data filtration

The IMU is read at a 100 Hz rate by an interrupt loop and all of the incoming data are filtered by a second-order Butterworth IIR² Low pass filter to attenuate high-frequency noise. Such noise may be caused by the sensor itself (on the least significant bits) and by the spinning propellers of the UAV. The propellers used on the UAV in this thesis vibrate at frequencies ranging from 57 to 97 Hz³.

The cutoff frequency of the filters is set to 10 Hz for the accelerometer and 20 Hz for the gyroscope. These values have been found experimentally and tested to work reliably. Results of the filtration can be seen in Figure 3.3 which shows the raw (blue) and filtered (red) accelerometer data in one axis.

²IIR (Infinite Impulse Response) filter is a recursive digital filter which output is computed by using the current and previous inputs and previous outputs [20].

³Information about motors and propellers used by the MRS group can be found here: https://ctu-mrs.github.io/docs/hardware/motor_tests.html

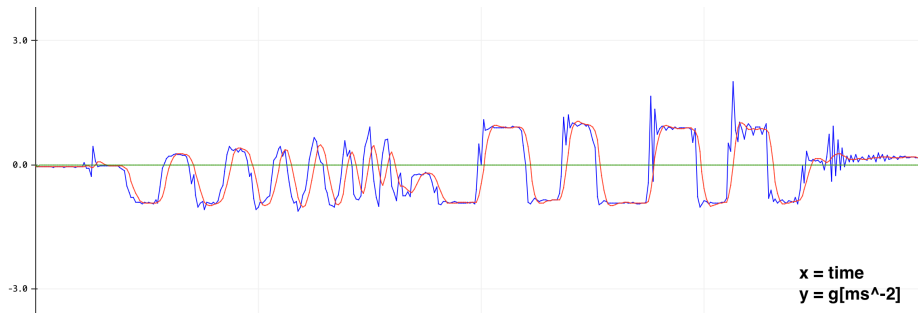


Figure 3.3: Raw and filtered X-axis accelerometer data

Both of the filters were designed using an online DSP (Digital Signal Processing) tool ⁴. Their structure and coefficients for accelerometer and gyroscope filtering can be found in figure 3.4 and table 3.1. The filtered data are then used to estimate the Roll and Pitch Angles of the manipulator.

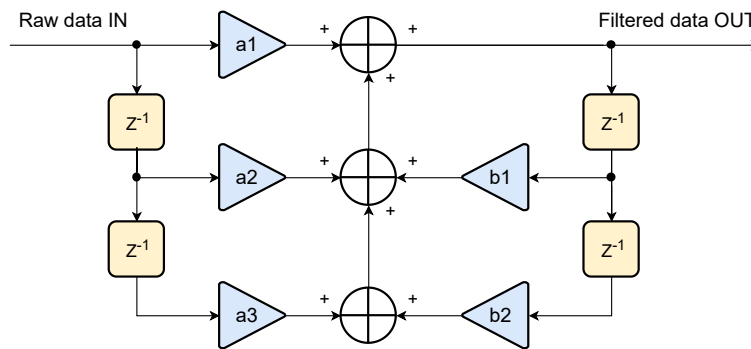


Figure 3.4: Structure of used 2nd order Butterworth IIR filters

coefficient	Accelerometer filter	Gyroscope filter
a1	0.067	0.207
a2	0.135	0.413
a3	0.067	0.207
b1	1.143	0.370
b2	-0.413	-0.196

Table 3.1: Coefficients used in digital filters mentioned in paragraph 3.1.1

⁴Micromodeler DSP tool available at <https://www.micromodeler.com/dsp/>

■ 3.1.2 Angle estimation

The roll ρ and pitch φ angles (according to figure 3.2) can be derived from the filtered IMU data as follows:

$$\begin{aligned}\rho_{accel} &= -atan2\left(\frac{\sqrt{A_Z^2 + A_X^2}}{A_X - \frac{\pi}{2}}\right) \\ \varphi_{accel} &= -atan2\left(\frac{A_Z}{A_X}\right) + \frac{\pi}{2}\end{aligned}\tag{3.1}$$

$$\begin{aligned}\rho_{gyro} &= \int G_X dG_X \\ \varphi_{gyro} &= \int G_Y dG_Y\end{aligned}\tag{3.2}$$

Where (A_X, A_Y, A_Z) are accelerations in all three IMU axes and (G_X, G_Y, G_Z) are angular velocities around all three IMU axes. Both methods shown in (3.1) and (3.2) should provide the same values but they do so at different qualities. Angles obtained from the gyroscope (3.2) provide a good measurement of quick changes but drift over time. On the other hand, angles obtained from an accelerometer (3.1) are stable in long term but very noisy for short-term measurements.

In both cases, this inaccuracy is caused by the sensor's noise. The gyroscope drift is due to the way the angle is computed from angular velocity (3.2) where the noise (which is not mathematically perfect with a mean value of 0) is integrated over time and causes the computed angle value to drift. The accelerometer is also naturally noisy, but the angle estimate does not use any integration of the noisy data (3.1). This makes the sensor unreliable for quick and small changes since the signal-to-noise ratio (SNR) is low for many accelerometers. But when used with a low pass filter which eliminates the noise (and quick and small useful data changes respectively) the sensor can then provide long-term accurate and stable output.

To get the best qualities of both measurements (both fast and stable long-term) a sensor fusion is used. To fuse the angles given by the gyroscope and

accelerometer a complementary filter ⁵ is used. The weights in the filter were, after several experiments, set to 95% gyroscope and 5% accelerometer (as seen in equation (3.3)).

$$\begin{aligned}\rho &= 0.95 \cdot (\rho_{prev} + G_X) + 0.05 \cdot \rho_{accel} \\ \varphi &= 0.95 \cdot (\varphi_{prev} + G_Y) + 0.05 \cdot \varphi_{accel}\end{aligned}\tag{3.3}$$

Note that ρ_{gyro} and φ_{gyro} from equation 3.2 are not used directly but integrated from G_X and G_Y in this equation (3.3) with the help of keeping previous values of ρ and φ (ρ_{prev} and φ_{prev}).

3.2 Kinematics

The kinematics section describes every mathematic aspect needed to move the manipulator. According to figure 3.1, the firmware has to compute an end effector positional difference caused by the UAV tilting, correct for it, and compute new joint coordinates to send to the servomotors - hence stabilize the end effector, in each loop. In parallel, the manipulator avoids collisions with itself and with the UAV's frame and receives commands for new positions from the UAV's computer. Schematically this process looks as follows:

⁵A complementary filter combines two inputs into one output in a way that the first input is multiplied by x (a number between 0 and 1) and the other is multiplied by $1 - x$. The products are then added to form an output. Complementary filters are widely used in sensor fusion algorithms.

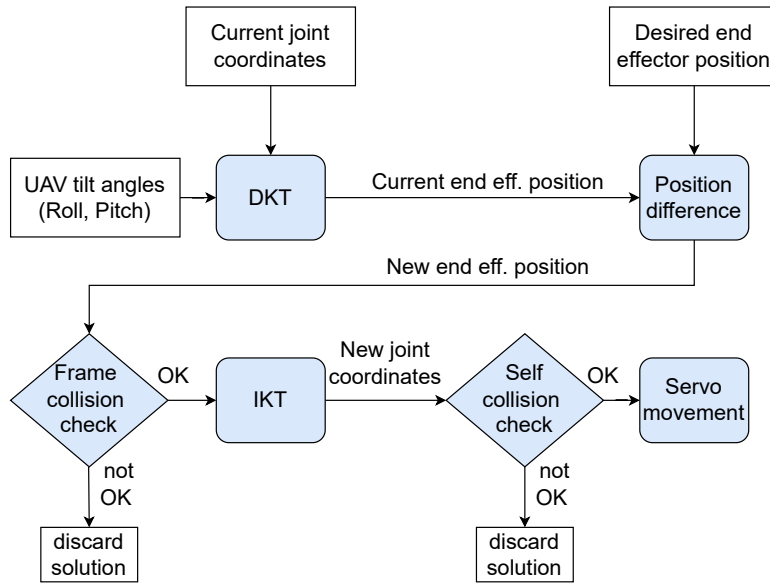


Figure 3.5: Stabilization algorithm process diagram

3.2.1 Denavit-Hartenberg parameters

Denavit-Hartenberg (DH) parameters are a set of numbers fully describing a robot's geometry. They can be derived for any open kinematic chain⁶ manipulator and are useful for creating transformation matrices for future kinematic operations [6]. The format of DH parameter vectors can be seen in (3.4)

$$\mathbf{DH}_i = [\Theta_i \quad \alpha_i \quad r_i \quad d_i] \quad (3.4)$$

where i is a joint number, Θ_i is the rotation angle around Z axis in the i -th joint (in case of rotational joints it is also a joint coordinate), α_i is the angle between i -th Z axis and $i - 1$ Z axis, r_i is i -th link length along the X axis and d_i is i -th position offset along the Z axis (or a joint coordinate in the case of prismatic joints). By these rules and drawing in Figure 3.6 it is possible to find all DH parameters for the manipulator built for this thesis (see equation 3.5).

⁶All links of the manipulator are connected in series through single joints (not forming any closed loops)

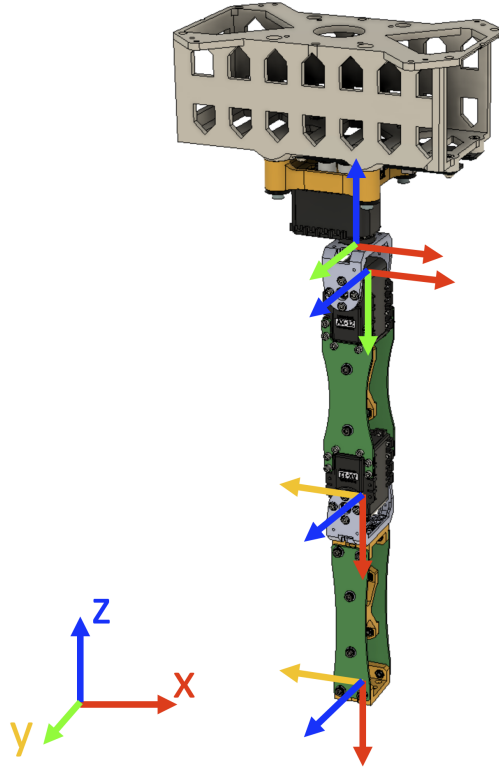


Figure 3.6: Manipulator drawing with highlighted axes in each joint

$$\begin{aligned}
 \mathbf{DH}_1 &= \begin{bmatrix} -\Theta_1 & \frac{\pi}{2} & 0 & -L_1 \end{bmatrix} \\
 \mathbf{DH}_2 &= \begin{bmatrix} -\frac{\pi}{2} + \Theta_2 & 0 & L_2 & 0 \end{bmatrix} \\
 \mathbf{DH}_3 &= \begin{bmatrix} \Theta_3 & 0 & L_3 & 0 \end{bmatrix}
 \end{aligned} \tag{3.5}$$

Where Θ_1 , Θ_2 , and Θ_3 are the manipulator's joint coordinates. Thanks to the derived DH parameters a set of transformation matrices can be made [7]. A transformation matrix \mathbb{T}_{i-1}^i is then used to transform a 3D vector pointing to $i - 1$ joint into a vector pointing to the i -th joint. The transformation matrix structure is shown in equation 3.6

$$\mathbb{T}_{i-1}^i = \begin{bmatrix} \cos\Theta_i & -\sin\Theta_i \cos\alpha_i & \sin\Theta_i \sin\alpha_i & r_i \cos\Theta_i \\ \sin\Theta_i & \cos\Theta_i \cos\alpha_i & -\cos\Theta_i \sin\alpha_i & r_i \sin\Theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

3.2.2 Direct kinematics task

Direct Kinematics Task (DKT) is a computation that takes a joint coordinates vector⁷ $\mathbf{J} = [\Theta_1 \ \Theta_2 \ \Theta_3]$ and all DH parameters (see 3.2.1) as an input and outputs a positional vector $\mathbf{X}_{ee} = [x \ y \ z \ 1]$ which is the extended⁸ end effector position ($[x \ y \ z]$) [9]. This whole operation can be performed by a matrix operation:

$$\mathbf{X}_{ee}^T = \mathbb{T}_0^3 \cdot \mathbf{0}^T = \mathbb{T}_0^1 \cdot \mathbb{T}_1^2 \cdot \mathbb{T}_2^3 \cdot \mathbf{0}^T \quad (3.7)$$

where $\mathbf{0}$ is an extended zero vector ($\mathbf{0} = [0 \ 0 \ 0 \ 1]$), which is then transposed, and \mathbb{T}_0^1 , \mathbb{T}_1^2 and \mathbb{T}_2^3 are transformation matrices made according to the rules stated in section 3.2.1. The computed end effector position can then be used in the stabilization algorithm (see section 3.5) and to detect collision between the manipulator and frame of the UAV (3.2.4).

3.2.3 Inverse kinematics task

Inverse Kinematics Task (IKT) is an inverse operation to DKT (3.2.2). The input of the IKT function is the desired end effector position vector \mathbf{X}_{ee} and the output is a set of joint coordinate vectors \mathbf{J} . The number of IKT solutions depends on a manipulator's construction. The manipulator made for this thesis always has four solutions (\mathbf{J}_1 , \mathbf{J}_2 , \mathbf{J}_3 and \mathbf{J}_4) all of which can be seen in figure 3.7.

⁷A vector containing all joint coordinates ordered from the base to the end effector.

⁸The position vector must be stated in homogeneous coordinates to allow it to be translated in 3D space and multiplied by a 4x4 transformation matrix (which contains the translation). The extension is done by adding a '1' as the fourth element of the 'xyz' position vector.

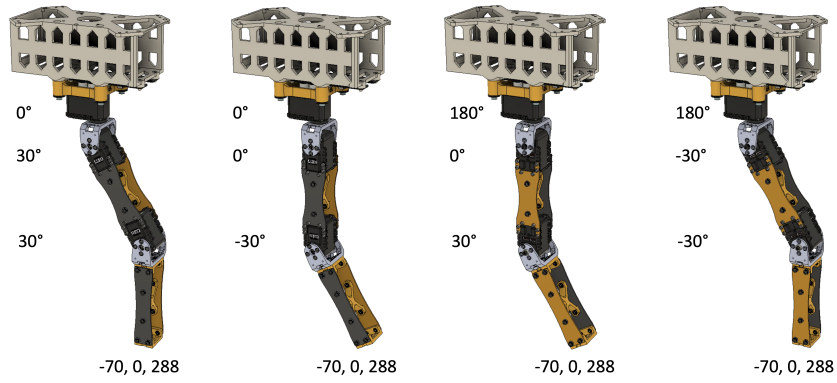


Figure 3.7: All four possible IKT solutions for the manipulator made for this thesis

■ Solving for Θ_1

Θ_1 (first joint coordinate, see figure 3.6) is trivial to compute from knowing the manipulator's origin and desired position. Since the manipulator's origin is set in the origin of the main coordinate system and the desired end effector position is given as an input ($\mathbf{X}_{ee} = [x_{ee} \ y_{ee} \ z_{ee}]$) then Θ_1 can be computed as shown in equation 3.8. Note that there are two valid solutions for Θ_1 .

$$\begin{aligned}\Theta_{1_1} &= -\text{atan2}\left(\frac{y_{ee}}{x_{ee}}\right) \\ \Theta_{1_2} &= -\text{atan2}\left(\frac{-y_{ee}}{-x_{ee}}\right)\end{aligned}\tag{3.8}$$

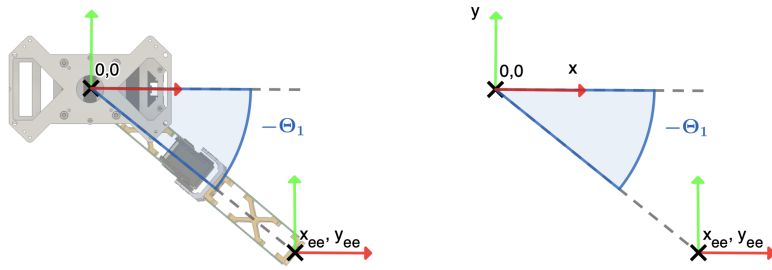


Figure 3.8: Topview diagram explaining computations shown in equations 3.8

■ Solving for Θ_2 and Θ_3

Due to the manipulator's construction, Θ_1 and Θ_2 are both constrained in one 2D plane which rotates around the main coordinate system origin Z axis based on the value of Θ_1 . This makes Θ_1 and Θ_2 independent of Θ_1 and allows for a 2D simplification of the problem (see figure 3.9).

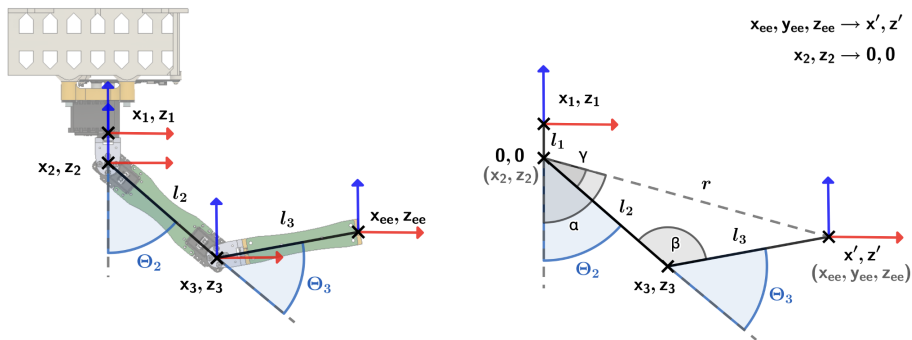


Figure 3.9: 2D view of Θ_1 and Θ_2 together with auxiliary parameters

To compute Θ_2 and Θ_3 several more auxiliary parameters are introduced. Their meaning is shown in figure 3.9 and computation in equations 3.9. Note

that all of the parameters are computed in two versions. One version for solutions 1 and 4 and the other version for solutions 2 and 3 are shown in figure 3.7.

$$\begin{aligned}
z'_1 &= z_{ee} + l_1 & z'_2 &= z_{ee} + l_1 \\
x'_1 &= \sqrt{x_{ee}^2 + y_{ee}^2} & x'_2 &= -\sqrt{x_{ee}^2 + y_{ee}^2} \\
r_1 &= \sqrt{x_1'^2 + z_1'^2} & r_2 &= \sqrt{x_2'^2 + z_2'^2} \\
\alpha_1 &= \operatorname{atan2}\left(\frac{z'_1}{x'_1}\right) & \alpha_2 &= \operatorname{atan2}\left(\frac{z'_2}{x'_2}\right) \\
\beta_1 &= \operatorname{acos}\left(\frac{r_1^2 + l_2^2 - l_3^2}{2 \cdot r_1 \cdot l_2}\right) & \beta_2 &= \operatorname{acos}\left(\frac{r_2^2 + l_2^2 - l_3^2}{2 \cdot r_2 \cdot l_2}\right) \\
\gamma_1 &= \operatorname{acos}\left(\frac{l_2^2 + l_3^2 - r_1^2}{2 \cdot l_2 \cdot l_3}\right) & \gamma_2 &= \operatorname{acos}\left(\frac{l_2^2 + l_3^2 - r_2^2}{2 \cdot l_2 \cdot l_3}\right)
\end{aligned} \tag{3.9}$$

Two versions of Θ_1 and Θ_2 can then be computed as

$$\begin{aligned}
\Theta_{2_1} &= \alpha_1 - \gamma_1 & \Theta_{2_2} &= \alpha_2 - \gamma_2 \\
\Theta_{3_1} &= \pi - \beta_1 & \Theta_{3_2} &= \pi - \beta_2
\end{aligned} \tag{3.10}$$

and the final four solutions (shown in figure 3.7) as this.

$$\mathbf{J}_1 = \begin{bmatrix} \Theta_{1_1} \\ \Theta_{2_1} \\ \Theta_{3_1} \end{bmatrix}, \mathbf{J}_2 = \begin{bmatrix} \Theta_{1_1} \\ \Theta_{2_2} \\ \Theta_{3_2} \end{bmatrix}, \mathbf{J}_3 = \begin{bmatrix} \Theta_{1_2} \\ \Theta_{2_1} \\ \Theta_{3_1} \end{bmatrix}, \mathbf{J}_4 = \begin{bmatrix} \Theta_{1_2} \\ \Theta_{2_2} \\ \Theta_{3_2} \end{bmatrix} \tag{3.11}$$

3.2.4 Choosing which IKT solution will be used

Given that there are four solutions to every inverse kinematics task it is needed to choose one of them to be used for the manipulator position. Firstly all of the computed solutions are checked if they are real numbers. In case of an unreachable position, the IKT solution results in a NaN (Not a Number value). The remaining solutions are checked for travel limits of each joint (they must fit into a range supported by each servo's range of motion). The remaining IKT solutions are then checked for collisions.

■ UAV frame collision check

To avoid collisions with the drone's body the end effector's z coordinate z_{ee} is compared to 0. If the z_{ee} is lower than 0 the position is valid, otherwise it is discarded.

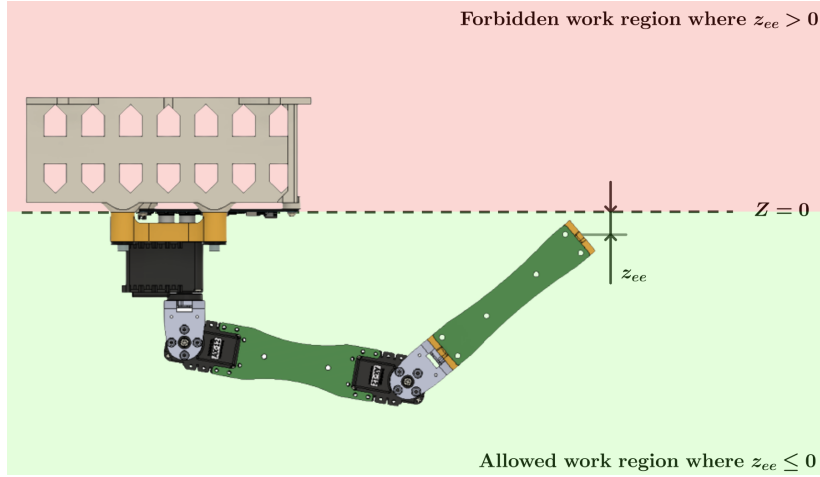


Figure 3.10: Visual representation of UAV's body collision detection

The next part of the UAV's frame are the landing legs. As can be seen in figure 3.11 there are four legs on the frame spaced 90° apart. The manipulator can collide with the legs only if Θ_1 (first joint coordinate) is $\pm 10^\circ$ around the leg angles ($45 \pm 10^\circ$, $-45 \pm 10^\circ$, $135 \pm 10^\circ$ or $-135 \pm 10^\circ$). If Θ_1 lies outside this region the position is valid. Otherwise, r and r_1 are computed as seen in equation 3.12 and checked if they are smaller than d (UAV's legs distance from the middle of the manipulator). This would mean that the manipulator cannot reach the leg, hence no collision occurs and the given position is valid.

$$\begin{aligned} r &= \sqrt{x_{ee}^2 + y_{ee}^2} \\ r_1 &= \sqrt{x_{J_3}^2 + y_{J_3}^2} \end{aligned} \quad (3.12)$$

In the equation above r is the distance between the manipulator's origin and the end effector in the XY plane. r_1 is the distance between the manipulator's

origin and the middle of the third joint in the XY plane. x_{ee} and y_{ee} are the end effector coordinates given in 3.2.3. x_{J_3} together with y_{J_3} make for the position of the third joint J_3 which can be obtained from equation 3.7.

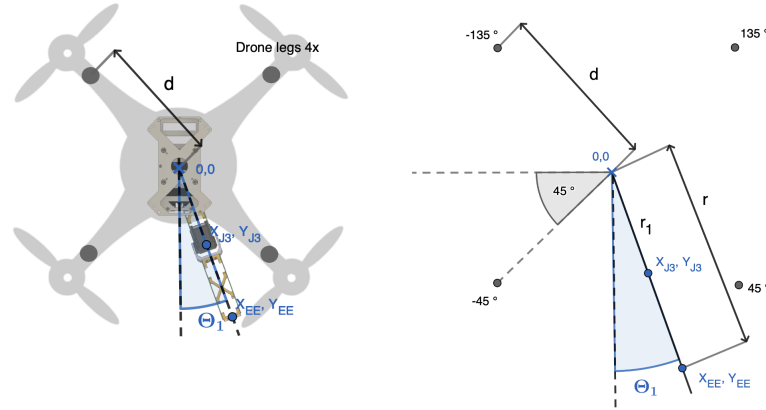


Figure 3.11: Visual representation of UAV's legs collision detection in XY plane.

■ Self collision check

If all of the found angles (Θ_1 to Θ_3) are within the travel limits of each joint the manipulator cannot collide with itself under any circumstances. This is caused by the construction of the manipulator which allows collision only in certain positions of joints 2 and 3 which are eliminated by the set range of motion for each joint.

■ Optimal solution selection

In the majority of cases, only one solution will remain valid after all collision and travel limit checks. If there are still more valid solutions the "easiest to perform" is chosen. The ease of performance is calculated as a Euclidian distance d_e between previous joint coordinates $\mathbf{J}_p = [\Theta_{1(p)} \ \Theta_{2(p)} \ \Theta_{3(p)}]$ and the new joint coordinates $\mathbf{J}_{1..4} = [\Theta_{1(1..4)} \ \Theta_{2(1..4)} \ \Theta_{3(1..4)}]$ (see equation 3.13). Whichever solution has the lowest distance (will move the manipulator's joints the least) is then chosen to be performed.

$$d_e = \sqrt{\Theta_{1(p)}^2 + \Theta_{1(1\dots4)}^2} + \sqrt{\Theta_{2(p)}^2 + \Theta_{2(1\dots4)}^2} + \sqrt{\Theta_{3(p)}^2 + \Theta_{3(1\dots4)}^2} \quad (3.13)$$

3.3 Servo communication

The Dynamixel AX-12A servos, used in the manipulator, communicate over a Half-duplex UART link. According to the AX-12A e-manual [11] the default communication parameters are:

- 1M Bd Baudrate⁹
- 1 Stop bit
- 1 Start bit
- 0 Parity bits
- Half-Duplex

The supplier recommends using an extra IC (Integrated Circuit) to convert the half-duplex signals from the servos to full-duplex Rx and Tx signals for the MCU (MicroController Unit). Since the used processor (STM32F411RE) can communicate in half-duplex mode directly then no additional circuitry is needed as far as the communications parameters are set correctly.

For sending instructions and receiving data from the servos the Dynamixel protocol 1.0 has to be used ([11]). In 2020 Ondřej Procházka in his Bachelor's thesis [12] created an STM32 library that can be used to conveniently communicate with the servos via simple instructions. The same library with minor changes is used for the servo communication in this thesis.

⁹By further testing of the finished manipulator, it was found that a slower baudrate of 200 kBd will be more suitable since the communication is more reliable.

3.4 ROS Communication

To control the manipulator from the UAV's onboard computer it needs to be able to communicate to ROS¹⁰. As described in section 2.2.2 the manipulator disposes with a USB interface directly connected to the onboard STM32F411RE. This enables a USB 2.0 Full-Speed connection between the manipulator and the UAV's computer.

On the software side, the whole communication pipeline from a ROS node demanding the manipulator's movement to the action itself, performed by the manipulator, can be seen in figure 3.12 and will be described in detail below.

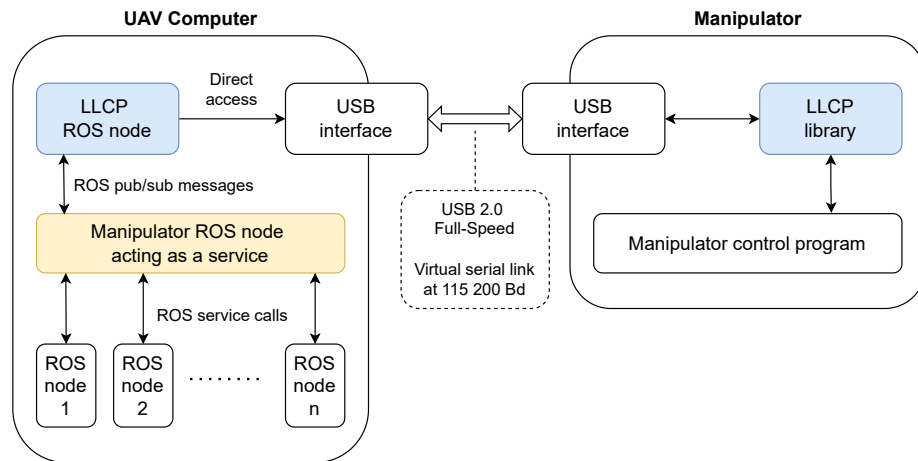


Figure 3.12: ROS communication diagram

3.4.1 Manipulator ROS node

A ROS node is a process that performs computation. Nodes are combined into a graph and communicate with one another using streaming topics (streaming topics being the edges and nodes being the vertices of the graph). The idea

¹⁰Robot Operating System (ROS) is a set of software libraries and tools that helps to build robot applications. It contains drivers, state-of-the-art algorithms, powerful developer tools, and is open source. More about ROS can be found here: <https://www.ros.org>

is that each node performs a simple computation and "calls" another node to perform another computation hence forming a complex system¹¹.

The manipulator ROS node (on the left side of figure 3.12) is the main in-ROS interface between other ROS nodes and the manipulator. It offers three services¹² that other nodes can call via ROS service calls to control the manipulator (see below). When a service is called, the node processes it and publishes a message addressed to the LLCP ROS node (left side of figure 3.12).

■ Enable service

The enable service accepts a Boolean argument and enables (or disables) the manipulator. When disabled, the Dynamixel servos are disconnected from power and cannot move. This makes the manipulator safe to touch and allows for having the control board powered on without the risk of the manipulator moving spontaneously.

When enabled, the Dynamixel servos receive power, slow speed is set and the manipulator moves to a parked position (see figure 3.13) and holds it until receiving next commands from the UAV. This allows the aircraft to land or takeoff safely without damaging itself or the manipulator.

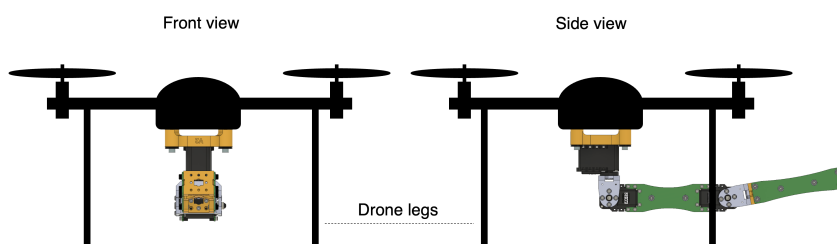


Figure 3.13: Manipulator set in the parked position.

¹¹More about ROS nodes can be found here: <https://wiki.ros.org/Nodes>

¹²A service is a request-reply type of communication inside ROS. More about ROS services can be found here: <https://wiki.ros.org/Services>

■ Engage service

The engage service accepts a Boolean argument and engages (or disengages) the manipulator and **only works when the manipulator is enabled** (see the Enable service above). When engaged the manipulator is set to working speeds and goes to a set position (see Set position service below) which it holds forever or until changed. At this point, the stabilization is also active so the position is held regardless of the UAV's tilt.

When disengaged the stabilization is turned off, slow speeds are set and the manipulator goes back to the parked position (see figure 3.13) which it holds until the next command.

■ Set position service

The set position service accepts a 3-dimensional float vector¹³ (*float[3]* array) which it then uses to set the *x*, *y* and *z* coordinate to the manipulators end effector. The service works regardless of the enable and engage state of the manipulator. This makes it possible to set the goal position at any time but the manipulator will hold it only when enabled and engaged.

■ 3.4.2 LLCP ROS node

LLCP (low-level communication protocol) ROS node is an MRS-created library that simplifies and standardizes communication with UART-based low-level devices.

Whenever a message is published to `/$UAV_NAME/llcp/send_message`¹⁴, the library takes its payload, prepares it for sending and sends it to a low-level device over a UART line (in the case of this thesis a virtual serial link on USB 2.0). If a low-level device (in the case of this thesis the manipulator)

¹³In the testing implementation an MRS Vec4 message format is used (four-dimensional float vector) and the fourth dimension is discarded. In further development, a Vec3 message type will be created for a clean implementation.

¹⁴More about LLCP inner workings and message structures can be found here: https://github.com/ctu-mrs/mrs_llcp_ros together with an example ROS implementation https://github.com/ctu-mrs/llcp_example

sends a message over UART, the library extracts the payload and publishes it on `/$UAV_NAME/llcp/received_message` ROS topic for other nodes to use.

■ 3.4.3 LLCP library

Similarly, an MRS LLCP library exists for low-level devices. It is written in plain c programming language and can be implemented directly on an STM32F411RE processor used in this thesis¹⁵. When running on the manipulator an Rx USB buffer is read in regular intervals and parsed by the LLCP library. The library extracts the payload, obtains the message type, and fills predefined structures (c struct) with data.

In case of data transmission a c struct is passed to the LLCP library, which serializes the data into a suitable format and sends them over the USB line (fills a USB Tx buffer which is automatically transmitted over the line).

■ 3.4.4 Manipulator control program

This part of the manipulator's firmware takes care of interpreting the incoming data sent by the UAV's computer. Mainly it ensures that the manipulator stays in the parked position (figure 3.13) when not engaged and that it does not move nor engage when not enabled. It also solves situations where a correct command sequence does not comply. For instance when the manipulator is running (enabled, engaged, and receiving a valid position) and is suddenly disabled. In this case, the power to the servos is immediately shut off and the engage flag is also deleted so the manipulator has to be engaged again after enabling it.

As the last part, every 500 ms the program sends a heartbeat message to the UAV containing the following information:

- Enable state [1/0]
- Engage state [1/0]

¹⁵More about LLCP for low-level devices and the code itself can be found here: https://github.com/ctu-mrs/mrs_llcp together with an Arduino example https://github.com/ctu-mrs/llcp_example/tree/master/arduino_example.

- Set position unreachable [1/0]
- Current set position [float[3] - x, y, z]
- Current joint coordinates [float[3] - $\Theta_1, \Theta_2, \Theta_3$]

3.5 Full movement pipeline

As described in the whole software chapter (3) the manipulator can stabilize its end effector in space, avoid collisions and receive commands from a UAV it is attached to. To finish the whole chapter a high-level diagram (figure 3.14) is presented. The diagram shows how a manipulator movement command passes through all of the blocks described in this chapter and results in a real action.

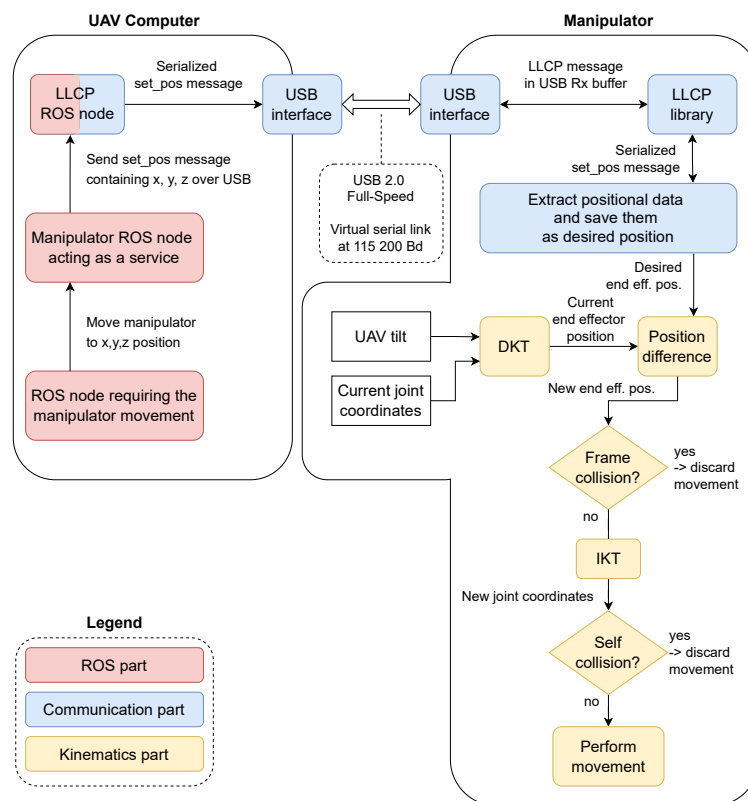


Figure 3.14: High-level movement pipeline diagram for better understanding of the software chapter as a whole.



Chapter 4

Implementation and testing

To get to the result shown below the whole solution was implemented in small steps exactly as described in chapters 2 and 3 and tested throughout. After ensuring the manipulator works electrically (see section 2.2.7) and is mechanically suitable for a real-world test (see section 2.1.5), several experiments on the bench were conducted.

Firstly the end effector stabilization was tested for fluency and accuracy. In the first tests, the manipulator was accurate but too unstable and shaky. After changing the maximum speeds and accelerations for the servos, together with speeding up the rate at which they receive instructions from the processor the situation got better. The final stabilization is illustrated by figures 4.1 and 4.2 where the battery box (drone) is tilted both front-to-end and left-to-right. Note the end effector holds the same position relative to the surrounding environment.

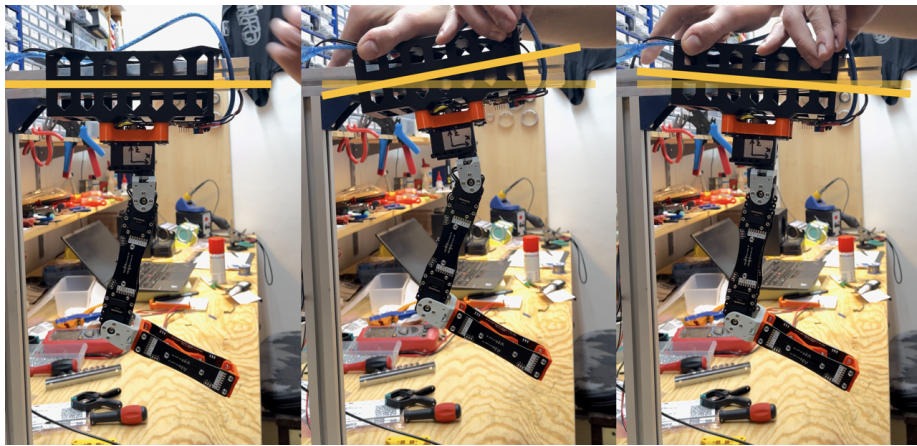


Figure 4.1: Front-to-end tilting illustration. [left = level, center = tilted back, right = tilted front]

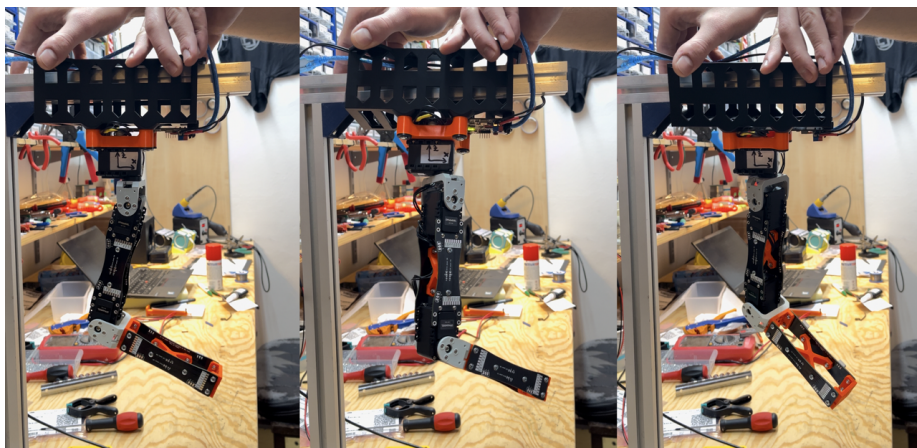


Figure 4.2: Left-to-right tilting illustration. [left = level, center = tilted left, right = tilted right]

After testing the stabilization several tests were made to see if the manipulator reliably communicates with the UAV and that it gets turned off every time the UAV sends a disabling command. This made the manipulator ready for in-flight testing.

4.0.1 In-flight testing

The manipulator was mounted to a Tarot T650 Sport¹ drone using 8 2.5 mm screws and the provided MRS battery box (the system can be seen in figures 4.3, 4.4, and 4.5 below). After handheld reliability testing the whole assembly was tested in flight which is shown in figure 4.6.

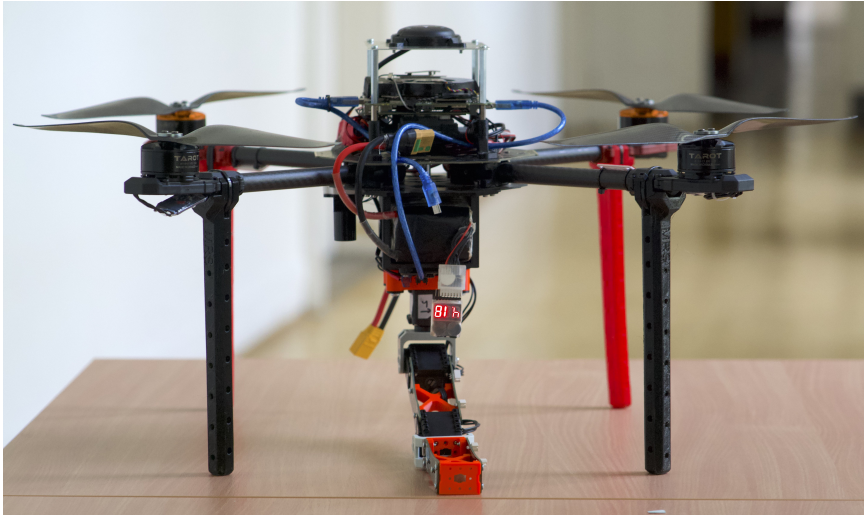


Figure 4.3: Front view of the manipulator attached to Tarot T650 Sport UAV



Figure 4.4: Side view of the manipulator attached to Tarot T650 Sport UAV

¹More about the MAVs used by the MRS group can be found here: <http://mrs.felk.cvut.cz/research/micro-aerial-vehicles>

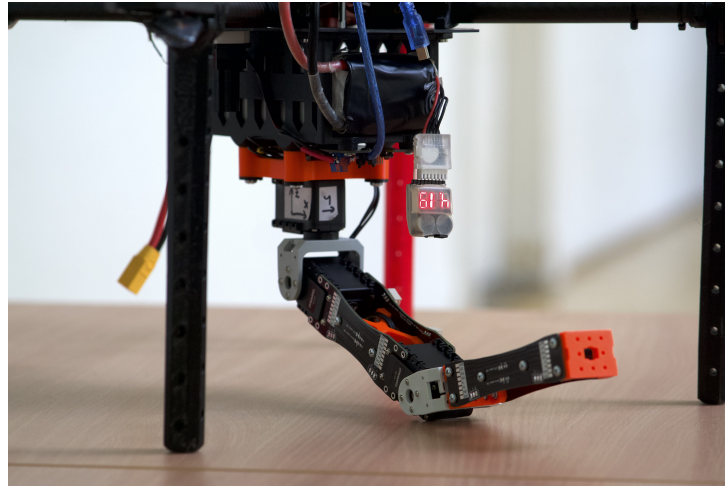


Figure 4.5: Detailed view of the manipulator attached to Tarot T650 Sport UAV



Figure 4.6: Flying UAV with the attached manipulator [left = camera mounted on UAV, right = view from ground]

The aerial test showed that the UAV can fly with the manipulator attached and that the manipulator can be safely operated by remote commands (see section 3.4.1). Another observation is the end effector stabilization not working correctly when in the air. This may be caused by additional accelerations in the XY plane generated by the drone changing its flight direction. Such behavior can interfere with the roll and pitch angles estimation (see section 3.1) hence changing the correction for the end effector position (see section 3.2) and making the manipulator behave incorrectly. A solution to this observation is proposed in section 5.1.



Chapter 5

Conclusion

The goal of this thesis was to design and build a lightweight robotic manipulator with a stabilized end effector to be attached to an unmanned aerial vehicle. This was achieved by designing a manipulator consisting of Dynamixel AX-12A servos, Dynamixel plastic brackets, 3D printed parts, and links made out of FR4 material used for PCB manufacturing (section 2.1). Together with the mechanics, a custom printed circuit board, containing the manipulator's electronics, was designed. The PCB contains multiple voltage regulators, a USB interface, an STM32F411RE processor, and an IMU, which allows for onboard end effector stabilization (section 2.2).

On the software side, firmware for the STM32 was developed that accepts commands from the drone and stabilizes the manipulator's end effector against the drone tilting in two axes (section 3). Together with the firmware a ROS node was developed to provide an interface between the MRS groups software and the manipulator (section 3.4.1).

The solution was then tested in flight in a real-world experiment where it proved to be reliable and capable of navigating in 3D space based on the UAV's commands. Results of the experiment also show that the stabilization algorithm implemented on the STM32 needs further development to become ready for in-flight usage (section 4). Additional testing with a functional gripper mounted to the end effector was planned but the tight schedule of the MRS group did not allow to perform these tests in time for this thesis.

5.1 Future work

Several points for future optimizations and enhancements arose during the manipulator development. First and the most important being the stabilization and drone tilt estimation algorithms, which are not yet ready for real-world utilization.

A solution to this would be implementing a Kalman filter instead of a complementary filter to obtain information about the drone's tilt. Further improvement could also be made by moving the IMU from the main PCB to the end effector (see figure 2.6 for reference) hence making it sense the end effector's position more precisely.

Another approach would be obtaining the tilt data from the drone's stabilization system to account for lateral accelerations. Also getting the drone's tilt corrections going to the motors and using them as a feed-forward branch of the end effector position regulator would help the manipulator to be more precise.

Mechanically the structure of the manipulator could also be changed to allow for roll and pitch compensation by just a single joint movement which would make the manipulator better react to faster tilt changes.



Bibliography

- [1] David Štych. *Robotic Manipulator for an Unmanned Aerial Vehicle*. České Vysoké Učení Technické v Praze, 2021. Available here: <https://dspace.cvut.cz/handle/10467/94737>.
- [2] Min Jun Kim , Konstantin Kondak, Christian Ott. *A Stabilizing Controller for Regulation of UAV With Manipulator*. IEEE ROBOTICS AND AUTOMATION LETTERS, VOL. 3, NO. 3, JULY 2018. Available here: <https://ieeexplore.ieee.org/abstract/document/8283724>.
- [3] S. Di Lucia, G. D. Tipaldi and W. Burgard. *Attitude Stabilization Control of an Aerial Manipulator using a Quaternion-Based Backstepping Approach*. Autonomous Intelligent Systems Group, University of Freiburg, Germany, 2015. Available here: <https://ieeexplore.ieee.org/abstract/document/7324191>.
- [4] C. E. Lin and S. -K. Yang. *Camera gimbal tracking from UAV flight control*. 2014 CACS International Automatic Control Conference, 2014. Available here: <https://ieeexplore.ieee.org/abstract/document/7097209>.
- [5] InvenSense Inc.. *MPU-6000 and MPU-6050 Product Specification Revision 3.4*. InvenSense Inc. 1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A., 2013. Available here: <https://www.farnell.com/datasheets/1788002.pdf>
- [6] Doc. RNDR. Adolf Karger, DrSc, Doc. RNDr. Marie Kargerová, CSc. *Základy robotiky a prostorové kinematiky*, page 220. Vydavatelství ČVUT, 2000
- [7] *FORWARD KINEMATICS: THE DENAVIT-HARTENBERG CONVENTION*, page 73. Available here:

<https://users.cs.duke.edu/~%7Ebrd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf>

- [8] Baca et al. *The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles*. J Intell Robot Syst 102, 26, 2021.
- [9] Doc. RNDR. Adolf Karger, DrSc, Doc. RNDR. Marie Kargerová, CSc. *Základy robotiky a prostorové kinematiky*, page 214. Vydavatelství ČVUT, 2000
- [10] Quigley, Morgan, et al. *ROS: an open-source Robot Operating System*. ICRA workshop on open source software. Vol. 3. No. 3.2, 2009.
- [11] *Dynamixel AX-12A e-manual*. ROBOTIS Co., Ltd., 2022. Available here: <https://manual.robotis.com/docs/en/dxl/ax/ax-12a/>
- [12] Ondřej Procházka. *Robotic Fire Extinguisher Mounted to an Unmanned Aerial Vehicle*. České Vysoké Učení Technické v Praze, 2020. Available here: <https://dspace.cvut.cz/handle/10467/87699>
- [13] Micrel, Incorporated. *MIC5239 Datasheet*. Micrel, Incorporated; 2007. Available here: <https://www.tme.eu/Document/fa8a43531ff1002ba06384e08cc241c7/MIC5239-DTE.pdf>
- [14] Intel Corporation. *Power Delivery Design Issues for Hi-Speed USB on Motherboards*, page 9. Intel Corporation; 2022.
- [15] Diotec Semiconductor AG. *SMD Schottky Barrier Rectifier Diodes SK12 ... SK115*. Diotec Semiconductor AG; 2022. Available here: <https://www.tme.eu/Document/289385b35b8c9ed87e63c8821fa5e0df/sk12.pdf>
- [16] ALPHA & OMEGA Semiconductor *AOZ1269QI-02 datasheet*. ALPHA & OMEGA Semiconductor; 2014. Available here: http://www.aosmd.com/res/data_sheets/AOZ1269QI-02.pdf
- [17] Diodes incorporated *AUTOMOTIVE-COMPLIANT, 40V, 3.5A SYNCHRONOUS BUCK WITH INTERNAL COMPENSATION*. Diodes incorporated; 2020. Available here: <https://www.diodes.com/assets/Datasheets/AP64352Q.pdf>
- [18] I. Batarseh, K. Siri and H. Lee *Investigation of the output droop characteristics of parallel-connected DC-DC converters*. Proceedings of 1994 Power Electronics Specialist Conference - PESC'94; 1994. Available here: <https://ieeexplore.ieee.org/abstract/document/373859>
- [19] STMicroelectronics *STM32F411RE datasheet*. STMicroelectronics; 2016. Available here: <https://www.st.com/resource/en/datasheet/stm32f411re.pdf>

- [20] Ian Grout. *Introduction to Digital Signal Processing*, strana xx -yy. Digital Systems Design with FPGAs and CPLDs, 2008. Available here: <https://www.sciencedirect.com/topics/engineering/iir-filter>
- [21] ALPHA & OMEGA Semiconductor *AOD4130/AOI4130 datasheet*. ALPHA & OMEGA Semiconductor; 2011. Available here: <https://www.tme.eu/Document/27930b9a28969be8f794bf882ea109ea/AOD4130.pdf>



Appendix A

Attached CD contents

kestrtom_MastersThesis.pdf – this thesis in *.pdf* format

kestrtom_KiCad_manipulator-main-PCB – KiCad project containing the main PCB described in section 2.2

kestrtom_KiCad_manipulator-PCB-links – KiCad project containing the PCB links for the manipulator construction

kestrtom_STEP-Model_manipulator-with-PCB-links] – 3D model of the manipulator in *.STEP* format

kestrtom_STM32-CubeIDE_manipulator-project – A CubeIDE project containing the STM32 firmware.

kestrtom_manipulator_photos – Photos of the manipulator attached to the UAV.

kestrtom_manipulator_test_video – Video from in-air testing of the manipulator.