Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA BIOMEDICA, ELETTRICA E DEI SISTEMI

Ciclo 34

Settore Concorsuale: 09/G1 - AUTOMATICA

Settore Scientifico disciplinare: ING-INF/04 - AUTOMATICA

DESIGN, DEVELOPMENT AND GUIDANCE OF THE AIRBORNE'S QUADROTOR

Presentata da: Bruno Strano

Coordinatore Dottorato

Prof. Michele Monaci

Supervisore

Prof. Lorenzo Marconi

Co-supervisore

Ing. Andrea Sala

Esame finale anno 2022

"The most exciting phrase to hear in science, the one that heralds the most discoveries, is not -Eureka!- but -That's funny...-"

Isaac Asimov

Abstract

The main goal of the Airborne project is to develop, at technology readiness level 8 (TRL8), a few selected robotic aerial technologies for quick localization of victims by avalanches by equipping drones with two forefront sensors used in SAR operations in case of avalanches, namely the ARVA and RECCO. This thesis focuses on the design, development, and guidance of the TRL8 quadrotor developed during the project. We present and describe the design method that allowed us to obtain an EMI shielded UAV capable of integrating both RECCO and ARVA sensors. Besides, is presented the avionics and power train design and building procedure in order to obtain a modular UAV frame that can be easily carried by rescuers and achieves all the performance benchmarks of the project. Additionally, in addition to the onboard algorithms, a multivariate regressive convolutional neural network whose goal is the localization of the ARVA signal is presented. On guidance, the automatic flight procedure is described, and the onboard waypoint generator algorithm is presented. The goal of this algorithm is the generation and execution of an automatic grid pattern without the need to know the map in advance and without the support of a control ground station (CGS). Moreover, we present an iterative trajectory planner that does not need pre-knowledge of the map and uses Bézier curves to address optimal, dynamically feasible, safe, and re-plannable trajectories. The goal is to develop a method that allows local and fast replannings in case of an obstacle pop up or if some waypoints change. This makes the novel planner suitable to be applied in SAR operations. The introduction of the final version of the quadrotor is supported by internal flight tests and field tests performed in real operative scenarios by the Club Alpino Italiano (CAI).

Abstract (Italiano)

L'obiettivo principale del progetto europeo Airborne è lo sviluppo, ad un TRL8, di tecnologie selezionate di robotica aerea applicate alla localizzazione di vittime di valanghe utilizzando sensoristica di consueto utilizzo in operazioni di ricerca e soccorso e valanga. I sensori in questione sono ARVA e RECCO. Questa tesi si sviluppa seguendo il design, la costruzione, e l'implementazione della logica di navigazione del quadrirotore a TRL8 sviluppato durante il progetto. Viene descritta la metodologia che ha portato allo sviluppo di un frame in lega di alluminio capace di effettuare schermatura elettromagnetica per procedere all'integrazione dei sensori ARVA e RECCO. Inoltre, viene descritto il design e la costruzione dell'avionica e gruppi motore al fine di ottenere una piattaforma modulare in grado di soddisfare tutte le specifiche di performance di progetto e di essere di comodo trasporto e utilizzo per piloti e operatori. Vengono inoltre presentati tutti gli algoritmi a bordo del quadrirotore, ed in particolare, l'implementazione di una rete neurale convoluzionale regressiva per la localizzazione del segnale ARVA. Vengono descritti gli algoritmi di guida autonoma che permettono di generare pattern noti senza preconoscenza della mappa e senza computer di terra (CGS). Inoltre, si presenta un generatore di traiettorie iterativo che non richiede mappe note e che utilizza curve di Bézier per la computazione di traiettorie ottimali, dinamicamente compatibili con il mezzo, sicure e ripianificabili. Lo scopo è lo sviluppo di un metodo che consenta veloci e soprattutto locali cambi di traiettoria nel caso di ostacoli improvvisi o ripianificazioni non previste. Questo rende l'algoritmo presentato adatto a diverse applicazioni di ricerca e soccorso. Infine, la versione finale del quadrirotore è supportata dalle prove sperimentali eseguite internamente e da prove di volo certificate eseguite dal CAI in ambienti operativi reali.

Acknowledgements

First of all, I'd like to thank my supervisor and co-supervisor Lorenzo Marconi and Andrea Sala for introducing me to the Airborne project, which has been a stimulating and challenging experience with an important moral purpose. I also would like to thank my colleague Michele Furci who shared opinions, discussions, advice, jokes, and experiments and helped to form me as an engineer. A big thanks to my family that always motivated me to keep on and whose attitude contributed to creating a place in which reforge hearth and mind, home. Last but not least, a special thanks go to my girlfriend that was always by my side and shared with me a piece of her heart.

Contents

Acronyms

UAV	**U**nmanned **A**herial **V**eichle
SAR	**S**earch **A**nd **R**escue
CAI	**C**lub **A**lpino **I**taliano
RTL	**R**eturn **T**o **L**and
CGS	**C**ontrol **G**round **S**tation
RC	**R**adio **C**ommander / **R**emote **C**ontroller
HITL	**H**ardware **I**n **T**he **L**oop
TRL	**T**echnology **R**eadiness **L**evel
ICT	**I**nformation **C**ommunication **T**echnologies
SNR	**S**ignal **N**oise **R**atio
GND	**G**rou**ND**
ADC	**A**nalog **D**igital **C**onverter
ESC	**E**lectronic **S**peed **C**ontroller
PWM	**P**ulse **W**idth **M**odulation
PCB	**P**rinted **C**ircuit **B**oard
PVC	**P**oly**V**inyl**C**hlorid
EEPROM	**E**lectrically **E**rasable **P**rogrammable **R**ead-Only **M**emory
PC	**P**ersonal **C**omputer
USB	**U**niversal **S**erial **B**us
LED	**L**ight **E**mitting **D**iode
DC	**D**irect **C**urrent
AC	**A**lternate **C**urrent

1 Introduction

In this introduction we presents the motivation behind this work of thesis and the structure of the latter.

1.1 Motivation

Nowadays, the number of experiences in alpine environments, like ski touring, and generally speaking, in non-urbanized locations, gained a boost in popularity. The downside is that this determined an increment in accidents that may happen during the activity in these locations. As an example, is not uncommon that among those accidents there is the falling of an avalanche. In fact, during season 2020/21 snow avalanches claimed about 130 lives in Europe. Professional search and rescue teams are deployed to arrive at the first emergency call. The timing is a critical factor because 80% of the victims survive when rescued in the first 15 minutes. This makes the post-avalanche search and rescue operations one of the most difficult and critical SAR operations. Moreover, the location of the victims is not an easy task because the detachment of an avalanche causes burials under a blanket of snow of variable thickness, covering a very large area. People who are run over are not visible to researchers as they are often buried under the snow. For this reason, a particular technology is used to locate the victim of avalanches based on the so-called Avalanche transceivers, more commonly called ARVA, or passive transponders based on the RECCO system. Both devices allow locating signals emitted by coupled devices worn by the victim of the accidents. Nonetheless, the research can be very difficult due to the uneven and unstable terrain that can slow down the research operations. It is worth stressing that, in those applications, tempestivity is a key factor because after a few minutes the buried victim can experience hypothermia and, eventually, death. The development of a system to improve the safeness and efficiency of this kind of operation is attracting a lot of attention [21] [46] [55], but the solutions still present problems that prevent reliable applications in real SAR operations. A functional and ad-hoc studied application has been developed in the H2020 European project Airborne (Aerial Robotic Technologies for Professional Search and Rescue) [51] coordinated by the University of Bologna.
Participation in this project constitutes the background and motivation of this thesis. One of the goals of Airborne is the development of a heterogeneous robotic platform for search and rescue missions in hostile environments, such as the Alpine scenario, in collaboration with X-LOG ELEKTRONIK GMBH [56] and RECCO [54],

which helps to quickly locate victims of avalanches. It is worth noticing that the same technology can also be applied in different scenarios than the alpine SAR, making the development of the Airborne platform valuable for SAR general operations. The platform is composed of at least: one motor group with propeller, one payload, one battery pack, one electronic regulator for handling the batteries, and the motor group. Besides, it comprehends the chassis integrating the payloads, the radio module to communicate with the CGS and the RC. In order to meet the demands of the project, the platform has been chosen to be a quadrotor. My group, composed of members of Aslatech company, and I were responsible for the development of the Airborne's quadrotor with a particular focus on the integration with the payloads ARVA and RECCO and the automatic flight. The focus of this thesis is thus on the development of the platform, the integration with payloads, and the path planning strategies.

1.2 Organization

In chapter 2 we present and describe the development of the UAV. Additionally, the chapter goes through some details of the Airborne project that motivated some choices of the design procedure. The concept of EMI shielding and modularity of the UAV are presented and explained. Moreover, the major tests done for the development of the quadrotor are reported. Finally, the final version of the quadrotor is described. In chapter 3 we present all the electronics and components of the quadrotor. A particular focus is put on the design and choice of the power train components since it greatly determines the performance of the final machine. The chapter also describes some tests done for selecting some components and evaluating the performance of some configurations. The design of the avionics is important for guaranteeing certain features and performance and plays a major role in the integration with the payloads. In chapter 4 we present the data handling process implemented in the Airborne's UAV. In particular, the handling of the payloads data is of great importance and is different based on the actual payload. The ARVA signal detection algorithm implemented on board is described. Additionally, an innovative ARVA signal detection algorithm using a neural network is reported. The algorithm is a supervised regressive convolutional neural network that, by taking as input the reading of a 3 axis ARVA antenna and the altitude of the quadrotor, is able to output the position of the transmitting beacon in the drone's body frame.
In chapter 5 the automatic flight strategies are described. In particular, the onboard automatic grid search algorithm is presented. This feature is of key importance during the execution of search operation with no pre-knowledge of the map and without the possibility to use CGS or precomputed planners. In fact, due to the nature of SAR operations in alpine environments, the flight pattern has to be decided on the spot and changed in real-time according to the shape of the search area, most of the time irregular. Moreover, it has to be automatic, leaving to the pilot only the

burden to localize the victims based on the readings of the payloads without worries about driving the UAV. Additionally, a trajectory planner based on high degrees Bézier curves is presented. This planner is able to guarantee high degree continuity, dynamics feasibility, and safety of the trajectory without the need to know the map in advance and allows one the replanning of local parts of the trajectory without the need to recompute it all. This makes the algorithm suitable to be applied in SAR operations where obstacles might pop up or the trajectory might be changed in real-time. Chapter 6 presents the internal tests done to validate the machine and the test done by CAI in near-real and real SAR scenarios. The CAI's test turned out to be of outstanding importance in evaluating the real performance of the UAV in field operations. In particular, they allowed us to highlight potential problems and to correct and improve different features. This chapter describes the usual tests done for evaluating the performance of the Airborne's UAV with the different payloads, giving a clear opinion in how this platform can be useful in SAR operations. Finally, in chapter 7 the conclusions of the work and the possible future applications and studies based on the years spent on the Airborne project are presented.

2 UAV Development

In this chapter, we present the Airborne quadrotor and the main concepts behind the development of a quadrotor for SAR operations. UAVs are recently under high investigation by rescue teams for SAR operations because of the efficiency in moving over difficult terrains and the reduced cost in performing the same operation respect other resources (i.e. helicopters). The focus will be on SAR in alpine and forested environments, although the concepts described in this work are true for general SAR operations. The main contribution of this chapter is the development of a novel method for the construction of a UAV, in particular a quadrotor, that can be used in SAR operation where the payload is sensible to EMI.

2.1 The application

The goal of AirBorne [51] is twofold. On one hand the objective is to develop at TRL8 a few selected robotic aerial technologies for professional SAR teams associations with a specific focus on quick localization persons buried by avalanches by leveraging the successful EU FP7 project SHERPA [55] in which specific application contexts have been characterized and products for SAR in the Alps have been developed at TRL5. On the other hand, the robotic technologies developed will then constitute the initial "technological seed" on which an innovative ICT technology service network will be created at the European level. The final goal is to feed professional SAR teams/associations with effective ICT technologies by creating a business opportunity and contributing to addressing several shortcomings of current operations. Actions to set up the network and to promote its proliferation are the further objective of AirBorne. It follows that the development of a TRL8 UAV is preparative for the prosecution of the project. Additionally, the UAV must satisfy the following requirements and main functionalities:

- Terrain-relative altitude measurement: the remote operator needs to fix the flight altitude at which the victim search should be performed. As consequence, the drones have to implement an altitude hold autopilot. Furthermore, to compensate for the inherent phase-lag of the control chain the altitude measurement should be anticipative (sensor rotates toward the movement direction). This feature becomes essential to avoid crashes with the terrain when operating at high distances in an environment with few reference points;

- Automated victim search: The UAV must be able to perform automatic search patterns without the need to use a precomputed map. In fact, in typical avalanche SAR scenarios, the research area is unknown. Additionally, due to the nature of the operation, and to simplify the critical task of the operator, the pattern must be started and executed without using any external devices other than the RC;
- Collaborative search: this functionality is based on the previous ones and needs a communication link among the drone fleet. The network needs to be fully connected with the CGS and the RC and save all the telemetry data onboard. The remote operator must always be able to check all the data taken during the flight;
- Payload Integration: The UAV must fully integrate the two payloads (ARVA and RECCO) so it needs to be able to load the payloads data in the network.

Given the nature of the application, there is the need to respect requirements based on encumbrance and flight performance. All the specification has been established in collaboration with CAI in order to reflect the real requirements in a SAR operation with a UAV. The major specifications are reported in table 2.1.

Specifications	
Functions	Value
Max Weight (No payload)	4Kg
Max Motor axes distances	500mm X 500mm
Foldable Frame	Yes
Max payload	1Kg
Payloads weight	Recco-ARVA
Max Speed	20m/s
Max Ascent Speed	5 m/s
Max Descent Speed	3 m/s
Max Service Ceiling Above Sea Level	3500m
Max Pitch Angle	35°
Max Angular Velocity	Pitch: 300°/s; Yaw: 150°/s
Max Wind Resistance	12m/s
Max Flight Time	25m
Operating temperature	-30°+60°

TABLE 2.1: Major specifications table

2.1.1 Payload ARVA

First developed in 1940 by a Swiss officer and then improved from 1966 to the present, the ARVA is a low-range receiver and transmitter pair used primarily in avalanche search operations. The physical principle is based on the emission and

reception of an EM signal at $457KHz$. The emitter generates an electromagnetic field whose intensity decreases with distance. The flux lines are similar to a dipole, denser near the poles. Based on the flux lines intensity and the orientation of the antenna is possible to locate the position of the transmitter, that is held by the buried person. In fact, the sensed intensity has its peak when the receiver antenna is parallel to the flux lines. Conversely, the intensity is minimum when the receiver antenna is perpendicular to the flux lines. More details about the working principle of the sensor can be found in Appendix C.

Search methods

For the sake of simplicity let's define d_{max} as the maximum distance at which the TX signal can be sensed. Conversely, let's define d_{min} as the maximum distance at which the TX signal can be sensed when RX and TX are orthogonally oriented.

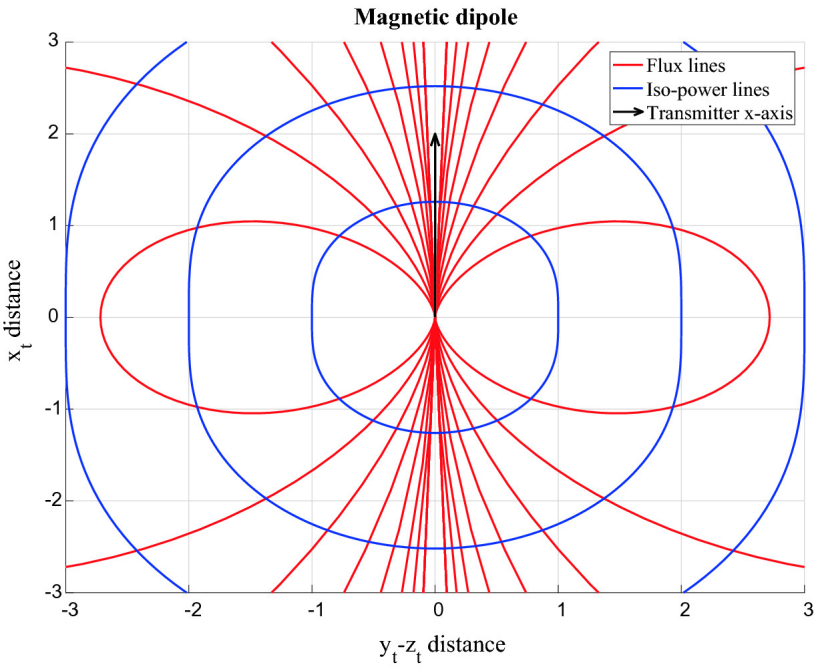


FIGURE 2.1: *Example of ARVA (dipole) flux lines. Their propagation motivates the different search methods.*

The search methodologies can be divided into the Cross method and the Directional method.

- The cross method converges to the TX following orthogonal flux lines. Moving in a straight line, and by keeping the orientation of the RX fixed, the signal received will increase to a maximum and then decrease. The same result can be obtained moving in a straight line orthogonal to the previous one. The process can be iterated and, progressively turning down the gain of the RX, is possible to converge to the position of the TX.

- The direction method converges to the TX by following the flux lines. In doing so, the rescuer usually performs a curvilinear trajectory but if the TX is buried vertically the trajectory will be linear. In order to follow the flux lines, the orientation of the RX must be continuously changed, searching for a maximum value of the signal. In particular, the RX is rotated by 120° in yaw respect to the z-axis of the rescuer.

In a real scenario, the described methods are used if the ARVA signal is sensed by the RX. Given a certain zone to be searched, the rescuer can use different patterns to search the area for signal detection. The shape used depends on the extension of the research area, the number of rescuers, and the condition of the terrain. The most used are the greek path, the triangular path, and the saw-tooth path. Is worth notice

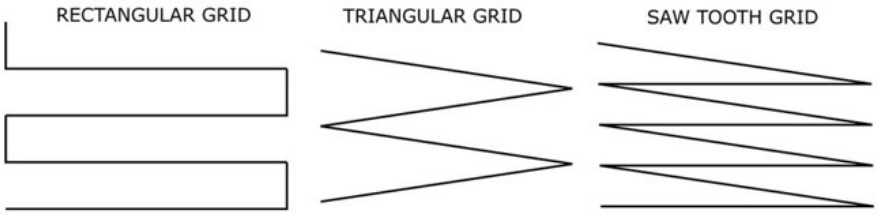


FIGURE 2.2: *Example of different types of grid executed during SAR operations*

these search paths are suitable both for search operations with human rescuers and with a UAV.

The commercial ARVA antenna cannot be integrated efficiently as a payload for a UAV. In fact, even if we manage to load the data into the UAV network, the antenna still needs to be rotated in yaw for the searching operation. This implies a systematic rotation in yaw of the drone, that can slow the operation, or the development of a gimbal system that turns the payload that complicates the design. This motivated, during the Airborne project, the production of an ad-hoc 3D receiver that can be loaded into the UAV network.

In order to avoid the chaotic behaviour of coupled antennas, there was the need to design a 3D antenna in such a way that neither antenna can influence the other. Since a magnetic loop antenna acts as a dipole, there exists a neutral plane. This is the plane of the windings. Hence, during Airborne, a globe-shaped loop antenna for three dimensions with three mutually independent loops has been constructed. To be able to measure the three dimensions of the 3D antenna simultaneously, three heterodyne receivers with a common local oscillator are required. The receivers mix the RF band of 457 kHz down to an intermediate frequency of approximately 10 kHz, which in turn can be sampled by the controller's ADC at 50 kSamples/s. The communication between the ARVA module and the drone is based on a fibre-optic link. At first, this avoids having a common GND, hence avoiding noise due to high-impedance or slightly floating GND potentials. At second, there is no link from

FIGURE 2.3: *3D version of the Arva Receiver developed during the Airborne Project.*

the drone to the ARVA unit coming close to the receiver or to the antennae. Links of conducting material could serve as guides for electromagnetic noise fields and thus decrease the receiver SNR. The 3D loop axis and the parallel receiver make this version of the ARVA suitable for UAV integration. In fact, the data coming from the receiver can be loaded into the UAV network and the 3 axis eliminates the need to tilt the antenna in yaw to search for signal detection.

2.1.2 Payload RECCO

Initially developed in 1973, RECCO is a system composed of a transmitter/receiver antenna, called *detector*, and the so-called *reflectors*, passive transponders composed of a diode and an antenna inserted in clothes, boots, and gears. The detector sends out a highly directional radar signal that is echoed back when it hits a reflector. The RECCO detector contains a transmitter on 866.9 MHz with an integrated Yagi antenna and a receiver on 1733.8 MHz with an associated Patch antenna. The searching principle of the RECCO System is based on the same principle as Harmonic Radar. The transmitter emits an 866.9 MHz signal that forms harmonic on 1773.8 MHz when echoed by a reflector. The receiver can detect the echoed signal and produces an audio signal based on the detection gain. Based on the tone, the rescuer can locate the buried victim. This is possible since both the transmitter and the receiver have a directional antenna, so the position of the reflector can be pinpointed. In order to avoid radio interference, the antenna transmits a pulse radio signal at 20% duty cycle and the RECCO system stays silent 80% of the time. The duty cycle is low enough for the modern digital-based radio system to not be affected by the RECCO signal. Besides, the operative regions when the system is used, and the search methodologies help to avoid the risk of radio interference with other devices.

Search methods

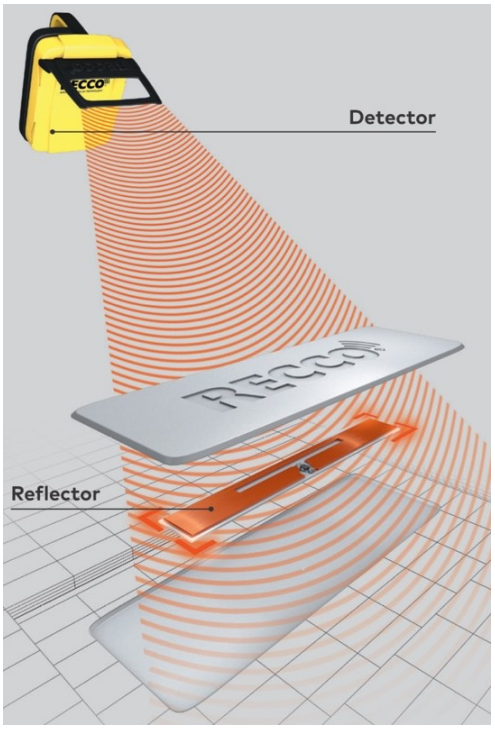


FIGURE 2.4: *RECCO system example*

In order to detect the first signal, the same path described in chapter 2.1.1 can be used. Once detected the first signal, search operations using the RECCO system aim to converge to the position of the reflector. The received signal gain decreases with distance. Instead, for a certain range, when the antenna is parallel with the reflector axis, the signal received is at maximum for the selected range. For this reason, the search methodology is based on side-to-side rotation to seek the maximum signal. The movement can be divided into three major steps:

- Sweep: The antenna is moved side to side for more or less 120° with respect to the initial position. This movement is used to detect the signal and to establish direction;
- Rotation: The antenna orientation is rotated 45° to the left side and 45° to the right side. The movement is used to detect the best signal, after the sweep;
- Turn: This movement is a slight turn of the antenna axis. The turn is used to re-aim the detector's signal cone back to the reflector when the reflected signal has been lost.

It is also possible to perform a fine search by progressively lowering the gain. The detector's area is a 3D diamond-shaped area. The signal is considered lost when the reflector is no more inside the signal cone. Is worth noticing that, the detection area are present, even if smaller, on the side and the back of the antenna. This forces the rescuer to not wear electronic devices when using the detector to avoid false detections. Attention is drawn to the fact that this could easily lead to problems when the

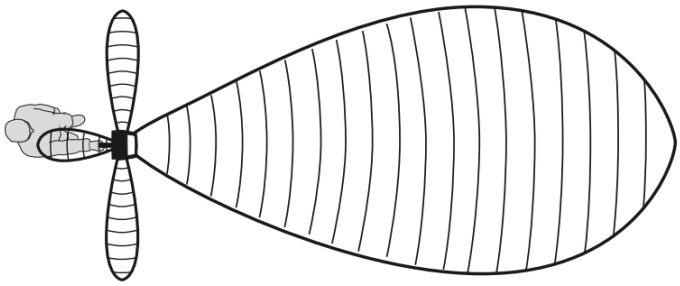


FIGURE 2.5: *Example of the detection areas of the RECCO antenna.*

antenna is mounted in a UAV.
In order to evaluate the entity of the problem, some internal test has been made. The component of the avionics and the power train group of the drone has been placed at a certain distance behind the antenna. The minimum distance to obtain a false detection has been recorded, for different antenna gains. From the tests emerged

FIGURE 2.6: *Testing of the minimum false detection distance with the battery pack of the power train group of the UAV.*

that the majority of components of the UAV act as reflectors, reflecting part of the signal at 1733.8 MHz. In particular, the cables turn out to be a huge source of interference. Some countermeasures can be taken: reduce the cable's length, reduce the physical contact between conductive parts, wrap the noisy component inside a shielding material, reducing the size of the backside lobe of the antenna. It follows that the commercial RECCO is not suitable for being integrated into a UAV. Besides, along the same reasoning did for the ARVA, even if we manage to load the RECCO

data into the UAV network without interference, the receiver still needs to be continuously moved following the three major steps previously described. This implies a continuous tilt of the UAV, which leads to a loss of performance. This motivated, during the Airborne project, the production of an ad-hoc multi-axis antenna that can be integrated into the UAV network and does not need to perform the 3 movements during the search routine.

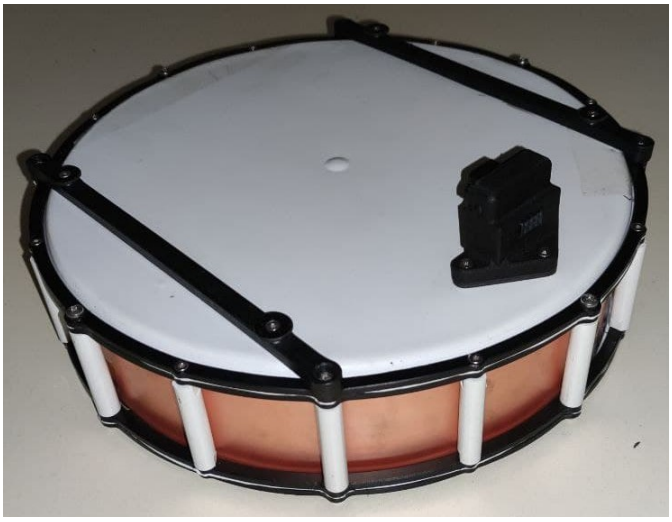


FIGURE 2.7: *The final RECCO module, inserted inside protection for the mechanical integration with the UAV.*

The new module presents circular polarization to detect the signal without the need to sweep, turn or rotate the module and the UAV. Besides, the backside lobe is reduced. In order to reduce the weight of the antenna, the new module is now powered by the UAV, which needs to supply a maximum of 8W and can be plugged directly into the 24V line using internal DC/DC to generate the +5V power line needed for the internal boards. Besides, the module is fully integrated into the UAV network through a serial port at 19200baud/rate.

2.2 Electro-Magnetic Interference

The integration with the two most used search and rescue technologies (i.e. ARVA and RECCO) is a big step forward in asserting the UAV as an efficient and usable machine for SAR. At the same time, is a very difficult task because of the interference that the UAV itself causes to the two payloads. Furthermore, due to the two different working principles and different working frequencies, the sources of EMI are different, making finding a method to shield the system, in both cases, a challenge.

2.2.1 First Shielding Tests

Since the Sherpa Project was noticed that electronic devices degrade the readings of the ARVA receivers. In order to take countermeasures, a full study of what was the

biggest source of EMI has been conducted. In a joint integration workshop between two Airborne's parters, Aslatech and X-Log Elektronik GmbH, held in Bologna, the impact of the drone's EMI on the ARVA receiver was analyzed and strategies to reduce this EMI were developed. In particular, in order to better characterize the EMI produced by the power train with battery, ESC, and motor, a dedicated test bench was implemented. An automatic procedure was set up by using an Arduino Pro Mini, which triggers the ESC with incremental steps of velocity, and Software developed in-house for reading and processing data and plotting based on Microsoft Excel VBA. A 2 axis test receiver, equipped with the same front-end that is intended

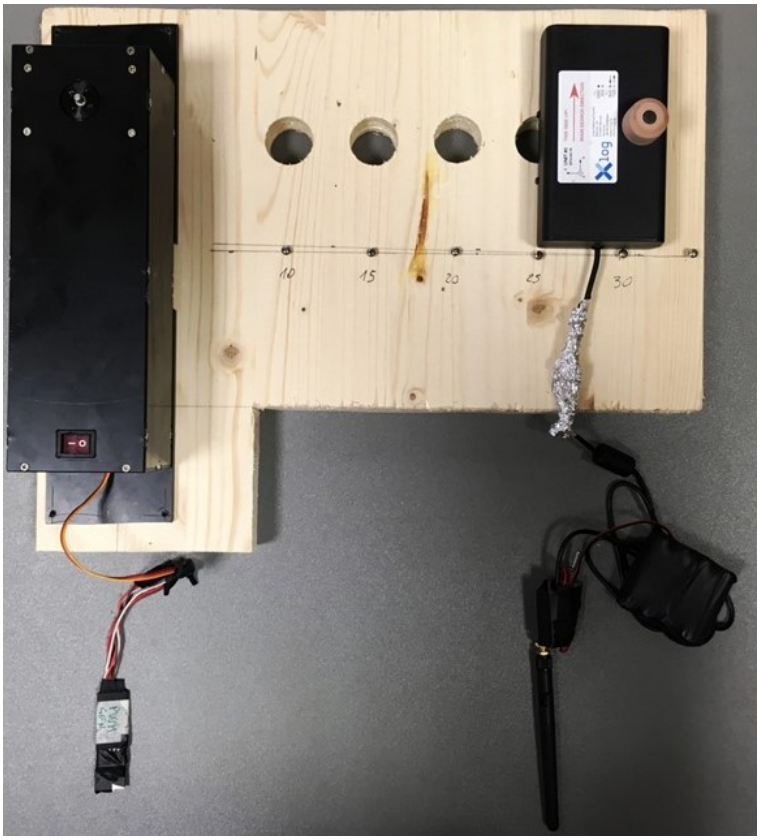


FIGURE 2.8: *The first test bench for EMI tests. On the left, there is the box that includes the motor, battery, and ESC. It is interfaced via a dedicated micro-controller that generates PWM signals for the ESC and at the same time sends serial data to a notebook. On the right, the ARVA that, via Xbee, transmits data to the notebook that are then paired to the PWM signal. The power train and ARVA are fixed into a wooden plate where the ARVA antenna can be moved at fixed distances from the motor (10, 15, 20, 25, 30 cm)*

to be used in the later 3D ARVA module, and a digital signal processing unit, that allows the analysis of EMI in various sub-bands of the signal band at 457 kHz, was used as ARVA antenna. The test showed, with repeatable results, that the major sources of EMI are:

- Switching components not directly related to the drone's power train, like DC/DC converters and digital logic;

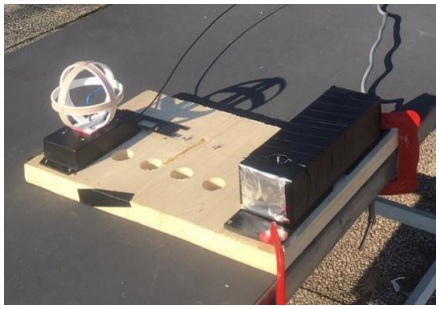
- The motor and the currents induced in the motor's coils;
- The motor controller and the high currents switching components;
- The behaviour of the battery under a high-frequency varying load.

During the tests, the engine remains in rotation for about 60 seconds and goes from 0 to 6000 rpm in steps of 100 rpm every 2 seconds. The tests have been repeated wrapping the power train in an aluminium shield. Here was experimentally verified that a simple electrically well-conducting shield could reduce noise caused by general switching electronics and high current components. In particular, full coverage of the entire electronics and a low impedance connection of all parts of the shielding is needed. Despite this, there was to need to determine a method to quantify the quality of the shielding and if it was sufficient to make the ARVA usable alongside a high EMI source like a UAV.
Consequently, the shielding capability has been further investigated using the new ARVA 3D antenna. The tests have been performed following a protocol derived from the one used previously. The protocol dictates the systematic combination of the following main variables:

- Shield presence;
- Distance between power-train group and antenna;
- Distance between antenna and transmitter;
- Presence of propellers.

The PWM signal that commands the turns of the motor is constantly incremented from a minimum of 1120 to a max of 1900 with a step of 20 every 2000 ms and again with steps of 1 every 500ms. Our aim was twofold: on the one side we aimed to find some spots in the PWM profile that generates less interference. If so, it would make it possible to control the UAV PWM profile in order to avoid noisy spots. On the other side, the objective was to characterize the EMI reduction caused by shielding different parts of the power train in order to find the best shielding strategy to be applied during the integration with the UAV.
The setup was composed by:

- ARTVA 3D module;
- Motor U3-KV700;
- ESC Alpha 10-LV;
- Propeller 13x4.4;
- Arduino Mega pro acquisition board;
- Aluminium based shielding materials.



In order to quantify how the EMI can disrupt the readings of the antenna, the background noise of the antenna has been measured and the readings of the antenna when the transmitter is in a different position have been recorded. In particular, the transmitter (i.e. the beacon) has been placed from 10m to 20m and at 30m away from the receiver in the 3 axis direction, aligning the principal axis of transmitter and receiver. The z-axis measured has been performed by tilting the antenna. This allows us to translate the reading of the antenna directly into a distance receiver transmitter, which is the variable of interest in SAR applications. The value read from the

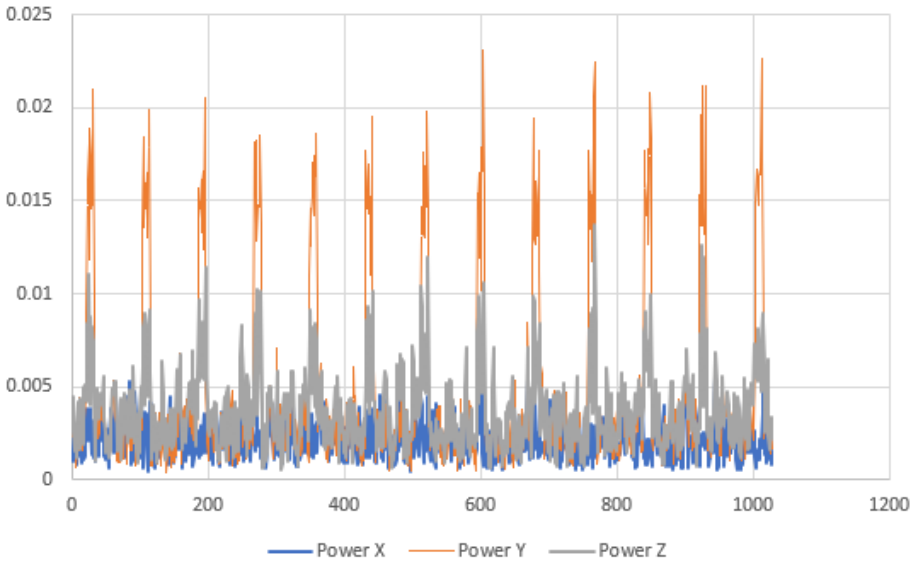


FIGURE 2.9: *Readings of the 3D antenna with the transmitter, aligned and placed at 20m along the Y-axis of the antenna*

ARVA module is a power-related measurement, coming from the log amplifier and analogic internal filtering.

Distance Antenna-Beacon	Value measured
10m	0.1
12m	0.09
14m	0.07
16m	0.06
18m	0.04
20m	0.02
30m	0.008
BG-Noise (No beacon)	0.007

TABLE 2.2: Reading of the ARVA antenna with transmitting beacon placed at different distances, with the principal axis aligned with the receiving antenna.

Tests Results

The results highlighted how much the choice of the material, and the shielding method, helps to reduce the EMI coming from the power train group, with both high current switching and low current switching components. As can be deduced from fig. 2.10a, the action of the aluminium shielding lowers significantly the EMI acting on the antenna. This is an important result because it traduces directly into having more range without the need to worry about false detection caused by the EMI. The results have been confirmed for all 3 axes of the module. Consequently, it follows that higher shielding capabilities turn into a bigger detection radius of the antenna. Is important to notice that the test has been arranged with only one motor,

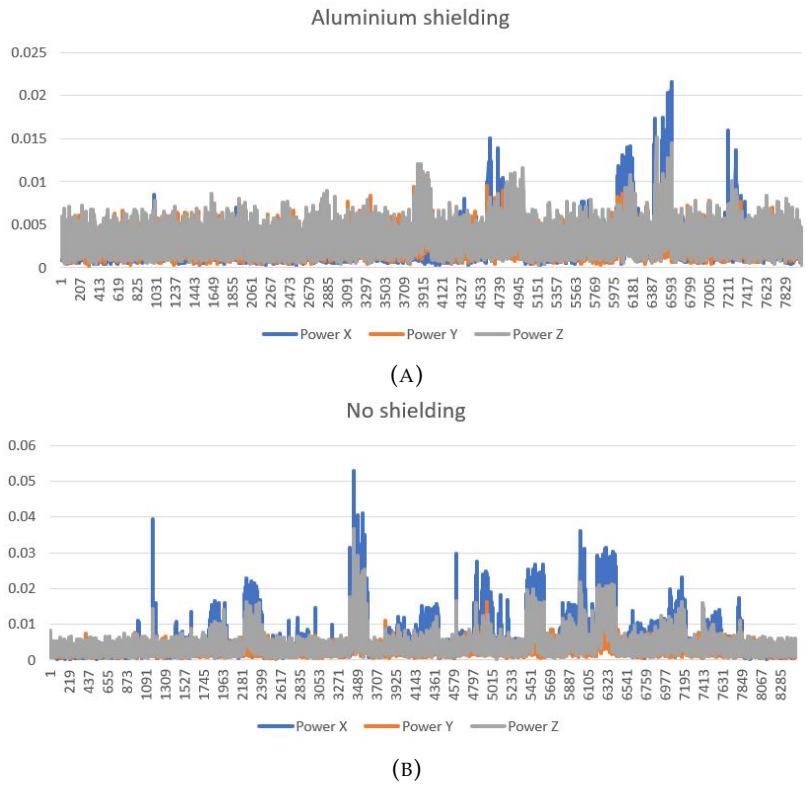


FIGURE 2.10: (A) *Readings coming from the ARVA module with the power train placed at 30 cm. An aluminium shielding, that covers all the power train group, is present* (B) *As for* (A) *but without shielding*

and the shielding, even if effective, was very primitive and impossible to be applied in a real scenario. Nonetheless, the results represent a turning point in finding countermeasures to fight the problem of EMI in the ARVA integration as a payload for a UAV.

2.2.2 UAV Shielding Frame

The aluminium, as a shielding material, provided a good solution for the EMI acting on the ARVA antenna. However, the poor physical proprieties of highly pure aluminium foils make it unsuitable to be applied in a real scenario, despite the good

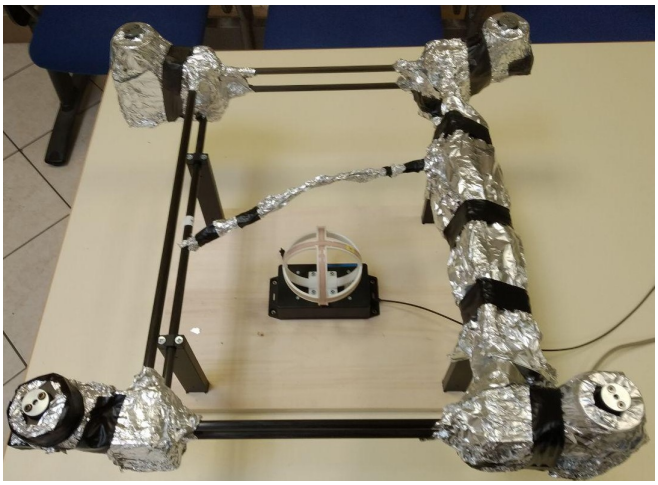
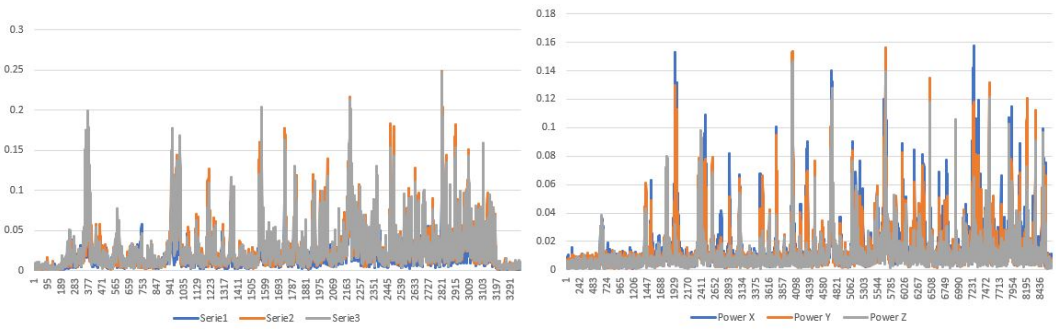


FIGURE 2.11: *The test ARVA bench with 4 shielded motor groups positioned in a quadrotor-like frame. The test aimed to prove if the shielding was effective in a situation similar to a real quadrotor application.*

shielding performance. Furthermore, even if the results obtained were encouraging, they were valid for just one motor group. Since the Airborne UAV will be a quadrotor, there is the need to extend the test campaign to a 4 motor group setup.
The first step is the creation of a frame structure in which the 4 motor groups, and the ARVA antenna, are positioned as if they were in the final UAV configuration. The new setup includes the same component of the previous tests. The firmware has been updated to handle and synchronize all 4 motors, based on the chosen PWM profile. Since the test bench aims to simulate the structure of a quadrotor, the wiring parts are present and shielded. The aluminium foils are used to shield every power group.

Results of Shielding with Aluminium Foils



(A) (B)

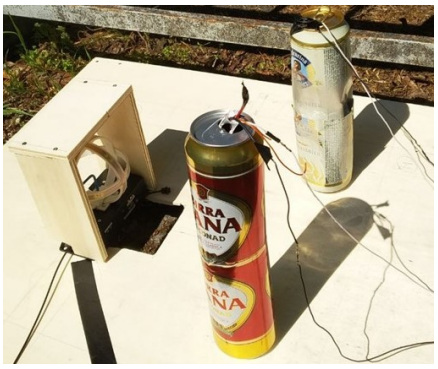
FIGURE 2.12: (A) *Readings coming from the ARVA module with quadrotor-shaped test bench with no shielding.* (B) *Readings coming from the ARVA module with a quadrotor-shaped test bench. Cables and power train groups are shielded with aluminium foils*

The results in fig 2.12a show, again, that without shielding it would be impossible to use the ARVA antenna because of the great interference that causes false detection

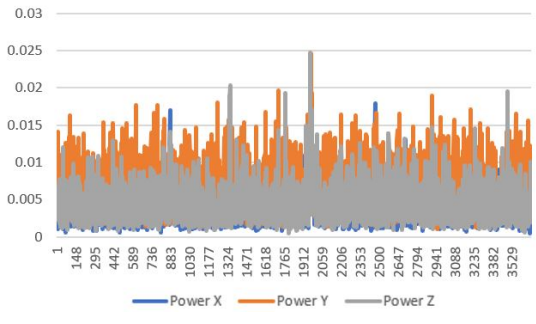
and may induce the rescuer to believe that there is a transmitter very close. The interesting result, which can be seen in fig. 2.12b, is that even with all the components shielded, the global performance is still not sufficient to integrate the ARVA as a UAV payload. In fact, with that magnitude of EMI, it would be very difficult to locate a victim in real operation, if not impossible. Successive studies make clear that the signal cables are a great source of EMI for the ARVA antenna. But even putting them far away from the antenna, the kind of shielding used seems to be not sufficient to obtain usable results in a configuration with 4 power train groups.

The New Shielding Method

During a successive test, we inserted the ARVA beacon inside an aluminium can. The results were very encouraging. In fact, even if the distance between transmitter and receiver was 1m the antenna was reading a signal from 10m. This motivates us to fit the power train group inside an aluminium can structure that also has better physical proprieties with respect to the shielding with aluminium foils. It was a decisive turning point inside the Airborne project because it was the first step to the development of a working Airborne UAV integrated with the ARVA antenna. The new setup included the presence of two power train groups, completely inserted inside two aluminium cans each. The cans have been sanded to eliminate the layer of polymer that was present. This allows the electrical contact between the two-part of the cans. The cans are placed in a wooden structure with the same distance, between the motor group and ARVA module, with respect to the previous test. The setup in fig. 2.13a produced excellent results in 2.13b. The test has been arranged following the same procedure described before. The extension of the test using 4 power train groups produced comparable results. The new shielding method turned out to be effective enough to be used as an EMI countermeasure in the development of the Airborne quadrotor.



(A)



(B)

FIGURE 2.13: (A)*Setup with two power train groups inserted inside two cans each.* (B) *Reading coming from the ARVA module for the test indicated in* (A)

2.2.3 RECCO Signal Protection

The EMI acting on the RECCO antenna is generated by the reflection of the transmitter signal, caused by the components of the avionics. The carrier signal is reflected into a signal with components at 1773.8 MHz, just like the action of a reflector. This causes false detections that could lead, as in the case of the ARVA, to think that there is a reflector nearby. Is worth stressing the fact that, this represents a huge problem in SAR operation because it reduces the efficiency of the operation by dilating the search times. In fact, in this kind of time-critical task, this can drastically decrease the surviving chances of the victim.
Conversely from the ARVA case, the action of the power train groups doesn't produce any EMI that could impact the readings of the RECCO module. This is true because the switching components and the EM fields, generated by the action of the DC motors, are at a much lower frequency with respect to the pass-bandwidth of the antenna. The result of the tests presented in chapter 2.1.2 suggested what are the major source of EMI for the RECCO payload. Despite, the new RECCO module, in fig.2.7, presents a reduced back lobe, countermeasures are still needed to eliminate the EMI problem. We arranged some tests aimed to study the optimal position of the avionics that produces the lesser interference. The results suggested that a distance of 30cm with respect to the centre of the quadrotor, is sufficient to avoid EMI if the payload is positioned right under the UAV. This motivated us to change the structure of the quadrotor, moving all the avionics on the side arms of the UAV and leaving in the centre just the payload. This idea will turn into the development of a modular UAV that is able to use both RECCO and ARVA systems and meets the transportability and ergonomics requirements of the Airborne project. Additionally, we developed a full set of PCBs in order to handle all the signals of the avionics without the need to have a huge amount of cables. In fact, the almost absence of cables plays a crucial role in reducing the EMI interference acting on the RECCO module.

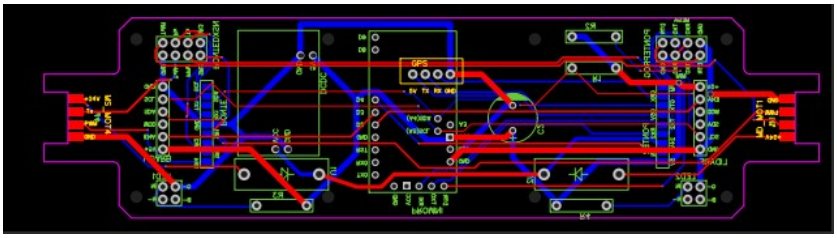


FIGURE 2.14: *The project of the first type of PCB. In particular, this was used to connect motor 1 and motor 4 and as an I/O to excess to the autopilot.*

2.3 Modularity

The production of a UAV for SAR operations imposes constraints in the development of the frame structure because, as indicated in the Airborne project specification in table 2.1, the quadrotor must be light and small enough to be carried by a human operator. These requirements, the need to insert the avionics and power train groups inside aluminium cans, and the need to put all the component away from the payload location has been fulfilled by designing a modular UAV. With the term modularity, we intend the ability to freely change some part of the machine without compromising the entire structure. The modules can be divided in:

- Power train module;
- Autopilot module;
- Internal logic module;
- Payload module;

Every module is plug-and-play. It follows that every module can be easily unmounted and a new one can be, just as easily, remounted. The modularity gives the Airborne quadrotor an important feature that can be extended beyond the SAR operation. In fact, if some module needs to be replaced we can perform the replacement without changing the other parts of the structure. Furthermore, this allows having plug-and-play payloads. The common mechanical and logical interface allows an

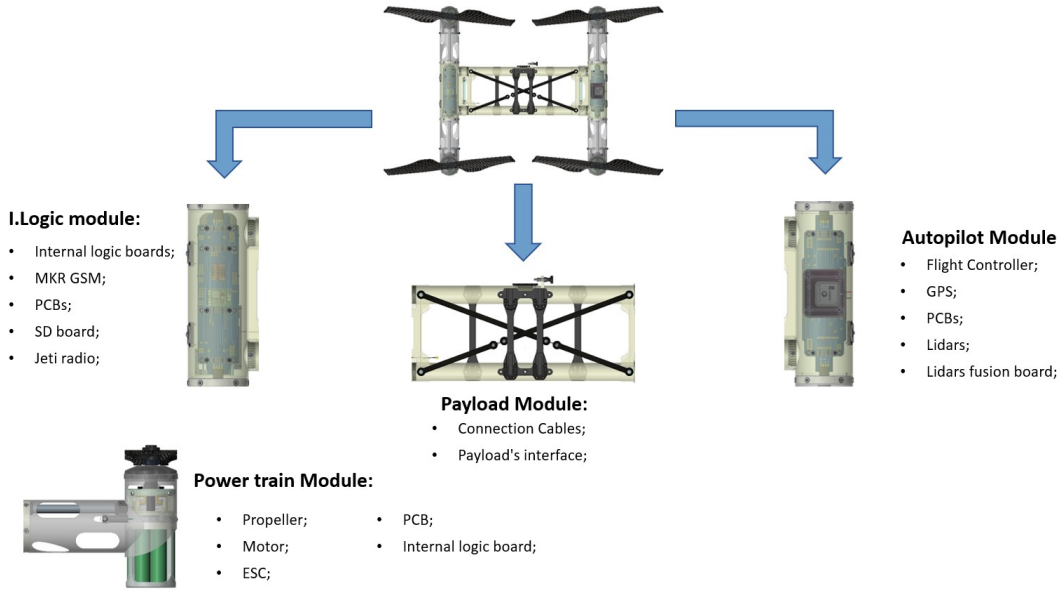


FIGURE 2.15: *Composition of the various module. Every module is designed to be connected with certain other modules. Furthermore, the design allows a quick replacement of every module.*

easy switch between payloads. In the particular case of the Airborne's quadrotor, the payload module was designed to be compatible with the ARVA antenna and the

RECCO antenna. However, the feature can be extended to other kinds of payloads (i.e. cameras and infrared cameras). All the functionality of the modules will be explained in chapter 3.4.

2.4 AlCan 02

The first version of the patented EMI-shielded Airborne's UAV is the *AlCan01*. The frame is composed of PVC and carbon fibre to maintain robustness and low weight. The central part is completely dedicated to the payload. In fact, not having components in the proximity of the payload reduced the interference both for RECCO and ARVA. The frame can be folded to make the UAV more comfortable to be carried when reaching the mission location. The cans, turned by Aslatech, are made of an

FIGURE 2.16: *The Airborne AlCan01 during a field test in an alpine environment using the ARVA payload.*

aluminium alloy that presents better physical properties and good electric conductivity, necessary to maintain the EMI shielding. They are connected to each other through shielded cables. Even if the various cans can be thought of as modules, the design is still not completely modular. In fact, changing a component without afflicting the others, even if possible, is not straightforward as it will be in the successive version of the UAV. The AlCan01 has been deeply tested in typical SAR-like scenarios by CAI. Their suggestions and tests had outstanding importance in leading us to design a machine that can flawlessly be used by a rescuer.
Although the generally good performances, there was the need to reinforce the frame against harsh meteorological conditions, which are typical during operations in alpine environments. Besides, the battery replacements needed to be improved because the whole replacement operation was relatively unconformable if performed with gloves. During the new design of the UAV, we decided to replace the PVC and carbon fibre parts with a full aluminium frame. The frame has been brazed and turned to give a smooth shape and maintain the necessary rigidity. Furthermore, it has been divided into 4 major modules, as described in chap. 2.3. The

FIGURE 2.17: *The Airborne AlCan02 during an internal flight test.*

UAV can be equipped with fixed pitch props or with foldable props, which helps to reduce the encumbrance during transportation. All the shielded cables that before connected the cans have been replaced with connectors built inside the modules. This allows a plug-and-play approach to replace a module, making the operation safer and quicker. Nonetheless, the ARVA and the RECCO modules were indeed usable with the AlCan01, an increase in shielding performance would lead to a bigger detection radius, and so to more efficient search performance. The further step in reducing the EMI was to reduce the dimension and number of all the cables that connect the modules by using, when possible, frame-integrated connectors. This will lead to less interference acting on the RECCO antenna and so having a RECCO detection ray bigger than 20 meters when flying at 20m of altitude. Furthermore, the electrical contact between the frame containing the avionics and the chassis of the motor groups helped to reduce the EMI acting on the ARVA antenna, increasing the radius to more than 20m when flying between 10m and 15m of altitude. Additionally, the power train groups have been re-designed in a more compact shape, reducing the total encumbrance of the machine. The AlCan02 is no more foldable, but it is possible to unmount the frame into 3 pieces that can be inserted and carried inside a backpack. This feature turns out to be of great importance when the UAV is operated in a zone accessible only through helicopters or cable cars. The mounting procedure is as simple as the dismounting one, the three pieces can be quickly mounted together and be ready to fly in less than a minute.

2.4.1 Payloads Integration

The payload module is designed to be compatible with both ARVA and RECCO. The mechanical integration allows a quick payload mounting thanks to a slider and

FIGURE 2.18: *The Airborne AlCan02 during an internal test with fixed pitch propellers.*

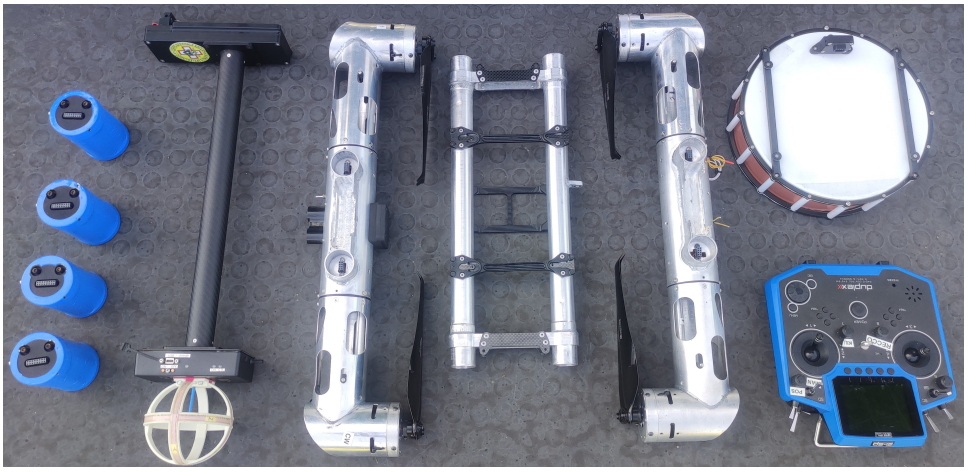


FIGURE 2.19: *The Airborne AlCan02 unmounted. All the pieces (three frame pieces, two payloads, one RC, four batteries) can be easily be carried inside a bag. The entire mounting procedure is simple and takes less than 1 minute*

a common interface. The internal software recognizes which payload is mounted and, consequently, transmits the data to the RC to be visible to the pilot. The entire procedure doesn't require screwdrivers or any similar tool to be accomplished, making it ideal for SAR applications. The RECCO module is inserted inside a PVC protection that: allows being integrated into the UAV, protects the internal components, is mounted right below the UAV payload module. The mounting position is strategical because the antenna has no obstacle in the frontal lobe and, at the same time, the avionics are outside the back lobe and the sides lobes.

The location of the ARVA module has been changed. Now the Antenna is placed on top of the UAV using the same mechanical integration used for the RECCO module. This choice is motivated by the fact that having the ARVA on top helps to keep the

(A) (B)

FIGURE 2.20: *The AlCan02 equipped with the RECCO module during a SAR operation in a real post avalanche scenario in* (A) *and equipped with the ARVA module during an internal test flight in via Felicina (BO) in* (B).

antenna safe during landing manoeuvres and, more importantly, can be distanced from the frame. In fact, the receiver is now on top of a carbon fibre tube, that contains the optical fibre inside, increasing the distance between the EMI generators and the antenna. The reduction of EMI acting on RECCO and ARVA allows to extend the detection radius, and so increases the search performance.

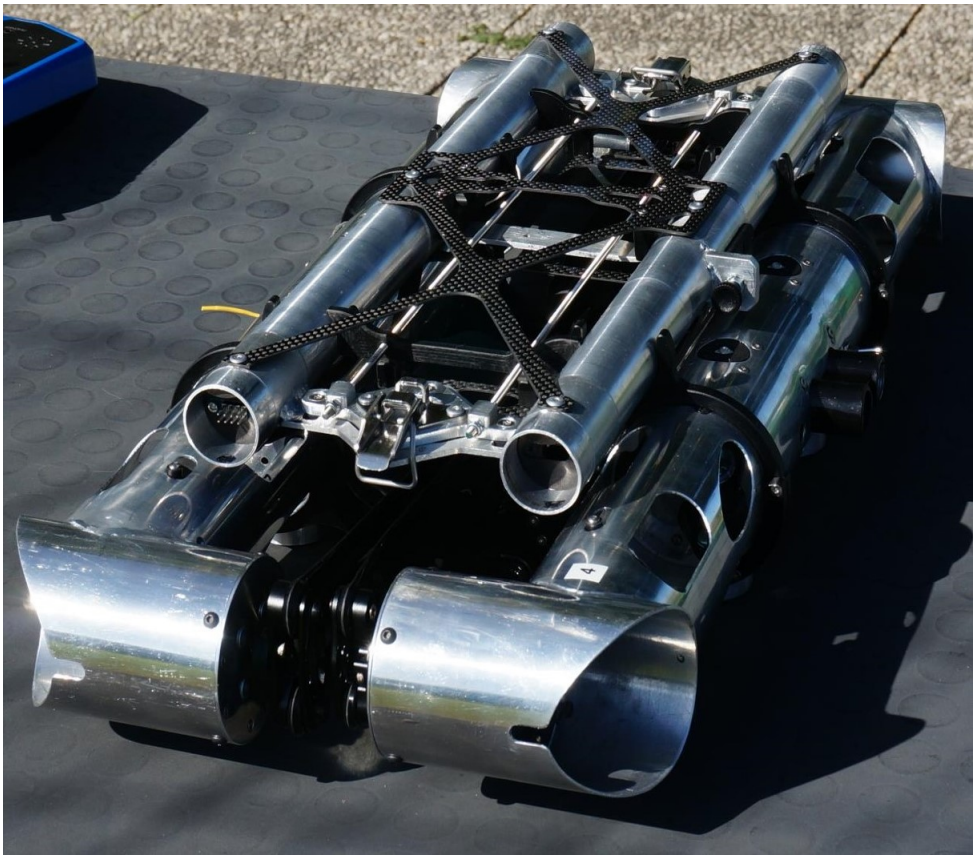


FIGURE 2.21: *The Airborne AlCan02 unmounted. The low dimensions of the drone when unmounted allow a rescuer to carry the whole quadrotor in a regular backpack. The foldable propellers help to reduce the encumbrance.*

3 Power Train and Avionics

Despite flight performance are greatly determined by the choice of the power train group, attention must be drawn to the fact that there are constraints represented by the EM emissions and by requirements to be respected for efficient SAR operations. In this chapter, we present the main methodologies and innovations in building the power train group and the avionics for a modular SAR UAV in application with EMI-sensible payloads.

3.1 Motor Test Bench

All the individual components of the power train group of the UAV has to be deeply tested and the performances verified, especially in a critical application like SAR. In order to execute reliable e repeatable tests, there is the need to create a test bench. During the Airborne project, we developed our internal automated test bench that was of paramount importance in the selection process of the components of the UAV. The main objective of the test bench are:

- Testing the performance of the selected motors;
- Testing the performance of the selected ESC;
- Testing the performance of the built battery pack;
- Testing the performance of the chosen propeller;
- Testing the EMI on the ARVA emitted by the power train group;
- Obtain the motor and batteries characteristic curves, in order to be able to estimate the UAV performance before going to flight;

The test bench components are:

- Arduino Mega Pro acquisition Board;
- Temperature sensor DS18B20;
- Strain Gauge load Cell;
- OpenScale acquisition board;
- Prop.- Motor - ESC - Battery pack.

FIGURE 3.1: *Motor test bench during an internal test session*

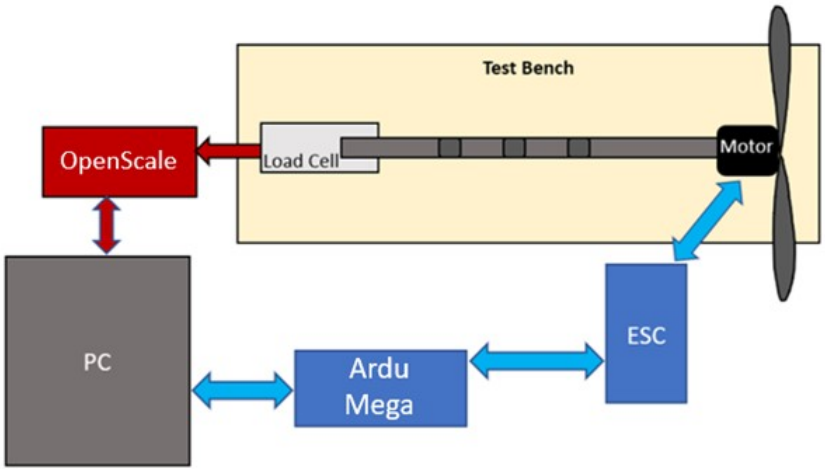


FIGURE 3.2: *The Airborne's test bench scheme. All the acquisition boards communicate with the PC using a proprietary shell that we developed.*

3.1.1 Load Cell

In order to measure the thrust of the motor, we integrated a load cell in the test bench. The load cell that has been used is a particular mechanical element that measures the effort applied thanks to the deformation of strain gauges attached to the body of the cell. The deformation causes variations in the electric resistance from which is possible to obtain the force applied to the cell.

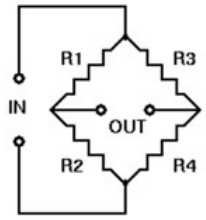


FIGURE 3.3: *The Wheatstone bridge scheme*

The ΔR is usually very small and is difficult and expensive to be measured. Consequently, it is better to convert that variation into something that is easy to measure. The Wheatstone bridge is able to convert the variation of electrical resistance in a variation of the output voltage. This method is cheaper and measuring the output voltage is easier. Substituting one of the resistances in the bridge scheme in fig. 3.3 with one strain gauge is possible to obtain an output voltage correlated with the variation of electrical resistance an input voltage following the formula:

$$V_{out} = \left(\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right) V_{in} \tag{3.1}$$

The acquisition board used for acquiring the load cell signal uses the amplifier HX711 and an Atmega328P programmable with Arduino. OpenScale comes with a native firmware useful to collect the signal coming from the load cell and for the execution of other functions. The firmware has been modified to be adapted to the particular test bench application. The load cell acquisition board communicates with the PC using a serial USB mini-B connection with a baud rate of $9600bps$. An internal EEPROM can be used to save and load our setup parameters every time the firmware is turned on.

Load Cell Calibration and Characterization

Despite the native firmware of the OpenScale board having a calibration procedure, in order to automatize the process we developed our calibration process. The calibration is needed in order to measure the loads correctly. Using the developed firmware, the user must insert a known weight in the load cell and the calibration gain will be changed until convergence. The precision is around 1 gram. Since the

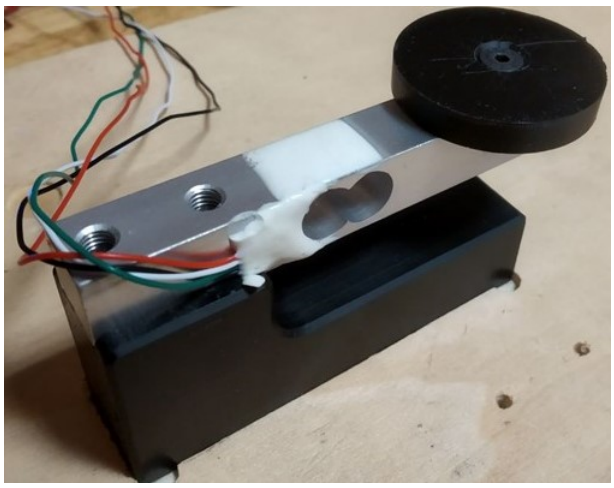


FIGURE 3.4: *Strain gauge load cell during the first calibration process.*

load cell has to measure the thrust of 1 motor, the scale has been tested for a range of measures starting from 0g to around 2,5Kg. Since there was no certificated weight

to be used, some object was measured by a commercial weight scale and used as reference. The commercial weight scale used is the Siena, produced by Soehnle, with a certified linear scale until 3Kg. Due to the shape of the particular load cell, in order to make the calibration repeatable and accurate, a circular platform has been attached to the load cell body. We used 20 different weights, with an incremental weight of around 13g, to calibrate the load cell. The procedure gave as result a linear scale in the range of interest.

3.1.2 Firmware

The automated test bench can be commanded via serial protocol from a PC. The developed shell allows visualizing all the data coming from the ESC and the OpenScale board. Furthermore, is possible to set the motor RPM through the PWM command, and to reproduce the PWM profile extracted from a log of a real flight. Every type of data is sent from the particular acquisition board to the shell through an internally-defined protocol. The communication between Mega Pro and ESC is a serial con-

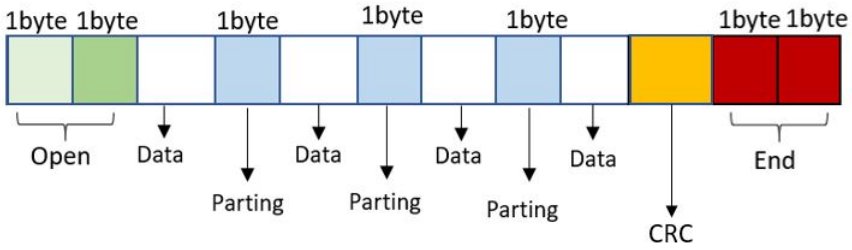


FIGURE 3.5: *Protocol used to communicate with the shell in the test bench. The number of data in the payload of the message may vary depending on the particular board.*

nection with a baud rate of *19200bps*. A signal handler function is implemented in the Mega Pro board that allows signalling the board with an interrupt in the RX pin when a new data pack is available. The protocol established by the producer of the ESC must be followed. In particular, for the ESC and motor tested, the protocol is composed of a frame of 26byte. As can be seen in table 3.1, the first 4 bytes are the header and the last 4 are used as a checksum. The shell is implemented in C# and allows the creation of an executable file with different kinds of functions that handle the data coming from both OpenScale and Mega Pro:

- **Cycle**: This function increments the value of the PWM sent to the motor to increment the number of RPMs for reading the data from the telemetry during the whole working range of the motor. To achieve this, a minimum and a maximum value for the PWM must be set together with the PWM increment step. For every step of the cycle, the data of the telemetry will be acquired and synchronized with the one coming from the load cell. Between the steps, there is a waiting time to allow the motor to reach a steady state. In this way is possible to avoid the acquisition of data during the transient phases;

ESC communication protocol				
Byte	Description	Type	Range	Measure
1	Frame Start	Unsigned char	0x9B	
2	Number of frame bytes	Unsigned char	0-255	
3	Protocol Version	Unsigned char	0-255	
4	Command word	Unsigned char	0x02	
5-6	Counter	Unsigned char	0-65535	
7-8	Throttle Input	Unsigned int	0-1024	%
9-10	Throttle Output	Unsigned int	0-1024	%
11-12	RPM(electrical)	Unsigned int	0-65535	
13-14	Motor Voltage	Unsigned int	0-65535	V
15-16	busbar Current	Signed int	$\pm$ 32767	A
17-18	Phase Current	Signed int	$\pm$ 32767	A
19	Temp MOS	char	0-255	
20	Temp C	char	0-255	
21-22	Motor Status	Unsigned int	0-66535	
23-24	Checksum	Unsigned int	0-65535	

TABLE 3.1: Alpha ESC communication protocol. This kind of ESC has been the most tested and used during the Airborne project

- **Single PWM**: is possible to send a single PWM value to the motor and acquire data for a particular condition;
- **Temperature measure (external)**: if an external temperature sensor is available is possible to read the temperature;
- **Tare to zero**: this function set the zero of the scale during the weight measures;
- **Arm Ratio**: To measure the correct value of the thrust coming from the motor is necessary to take into account the arms of the test bench. The ratio between the lever arms can be set with this parameter;
- **Average measure amount**: This function allows the selection of the number of samples to perform the mean value during the reading phase of the load cell in the OpenScale;
- **Measure Unit**: this function allows the user to select the preferred measure unit;
- **Calibration**: automatic calibration function;
- **Reset**: a function that reset the motor speed to 0 and deletes the data in the tables;
- **Stop Stop Smooth**: functions that will stop the motor. If the motors are powered by an external generator the function Stop Smooth allows a reduction of the PWM to zero, avoiding voltage spikes that can damage the generator;

- **Exports**: functions that create a text file with the whole data coming from the tests or log in a file just the data visualized in the current test;
- **Imports**: Functions that allow to import of a PWM profile or a load profile and regulate accordingly the RPM of the motor. The PWM sent to the motor are incremented accordingly the reduction in voltage, if powered with batteries;
- **Load control**: Function that allows setting track a certain reference thrust value.

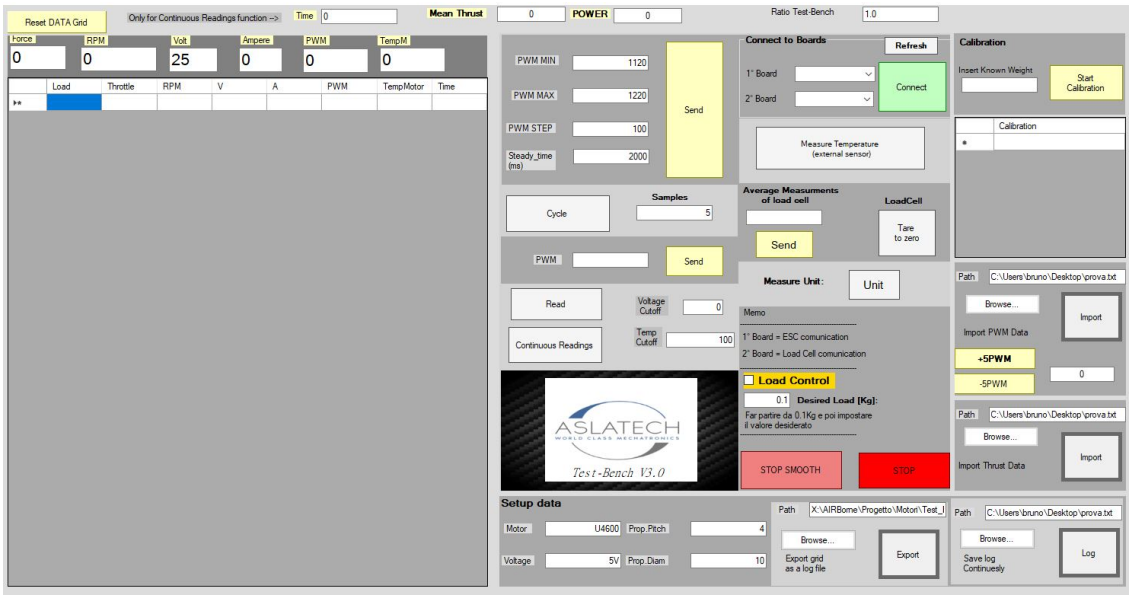


FIGURE 3.6: *Example of the test bench's shell appearance.*

3.2 Power Train

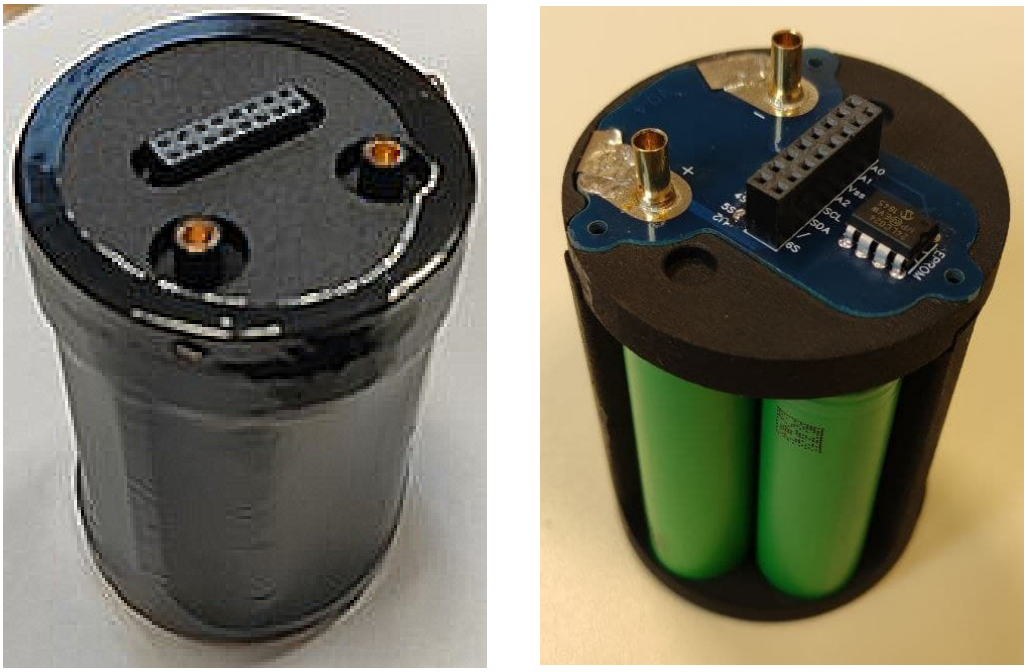
The selection of the power train components is a process that takes into consideration several aspects related to the final application in which the components will be used. In the Airborne case, the components need to be performance enough to let the UAV full-field the requirements of the project in table 2.1. The selection processes can be divided into three major test procedures: Power train characterization, EMI test, Temperature test.

3.2.1 Power Train Characterization

In selecting the power train components we did an intensive job in extracting the characterization curve of the motor and the discharge curve of the batteries under different loads. The first step is the selection of the components based on the performance reported in the datasheet. The further step is to intensively test the components combined or applied in a particular configuration, in order to check if the final results full-field the requirements. The tests performed are repeatable and follow an internal protocol in order to give results that are comparable with previous tests.

Battery Pack

Given the wanted flight time, the estimated weight of the quadrotor and the possible combination of motors we decided to test the Lithium-ion rechargeable cell INR18650. Those cells present a nominal discharge capacity of 2500mAh with a nominal voltage of 3.6V and a weight of 45.5g. Given the modular structure of the quadrotor, we need to power separately every motor group. Consequently, we need 4 battery packs. The single battery pack consist of 6 Li-Ion cells in series, an EEPROM to store the battery data, a digital temperature sensor DS18B20 and the connectors for the power and the logic. The connectors and the battery CAD are designed to offer a unique interface to both the drone and the battery charger. The battery is



(A) (B)

FIGURE 3.7: *The complete battery pack is used by the Airborne's quadrotor. In* (A) *the version with shielding and complete interface. In* (B) *the detail on the cells and the internal PCB with the EEPROM to store all the data.*

initially tested on the test bench. Given a certain load, that corresponds to the thrust generated by the motors during a hovering manoeuvre of the selected UAV, the discharge curve is obtained. If the results are above the expectations, a similar test is repeated using the UAV. In detail, 4 battery packs are tested during a real hovering manoeuvre of the UAV. Is important to notice that all the data coming from the batteries allows evaluating the performance even during standard flight tests. Since a fault can lead to a crash of the machine, they are always monitored. The performance of the cells INR18650 was not sufficient during the real application, because of the huge drop in voltage under load and the great loss of performance at low temperatures. We decided to use the Sony US18650VTC6, a higher density cell with the same technology Li-Ion. The nominal discharge capacity is 3000mAh, the weight is

46.6g, the nominal voltage is 3.6V. The 6s battery pack using these cells presents a smaller underload drop of voltage, between 1 and 1.5V, and great low-temperature performance. In fact, they were tested by CAI even without a pre-warm procedure.

Electronic Speed Controller

The ESC is an electronic circuit that mainly controls and regulates the speed of an electric motor through a network of field-effect transistors. Additionally, it can also collect data from the motor and communicate it to an external device. The choice of the ESC has to be paired with the choice of the motor and the maximum current. In fact, different types of electric motors require different types of ESC. In the Airborne project, there was an additional factor to be considered, the EMI. In fact, the switching rate of the ESC can influence the reading of the ARVA antenna but we registered no influence on the RECCO antenna. In fact, different ESCs afflict differently the ARVA antenna. Considering this additional factor we selected the Alpha-ESC 40A LV with modified firmware, made by T-motor, that allows a switching frequency, and its main harmonics, to be far from 457KHz. This helped to reduce the interference caused just by the action of the ESC. In fig. 3.8 can be clearly seen that the action of the ESC can be sensed by the ARVA antenna. In particular, in this case, the ESC is periodically searching for an input command and the ARVA reads as if there were a beacon between 15m and 20m.

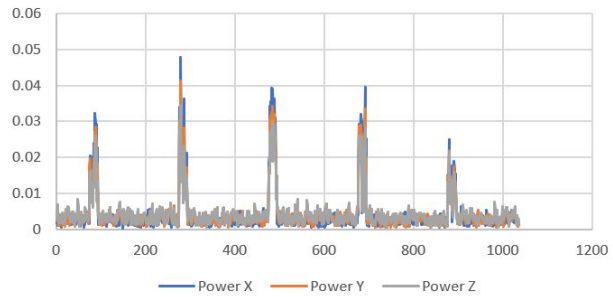


FIGURE 3.8: *The ARVA readings with no shielded ESC. The ESC has been turned on without giving a PWM input. Consequently, the regulator searches periodically for an input command.*

DC Motors

The motors evaluated are AC-brushless motors compatible with 6-S batteries. In the Airborne project, we had the additional constraint, of selecting components that produce low EMI, that fixed our ESC choice. This traduces in selecting the motors that have enough performance, to fulfil the project specification, and that is compatible with the Alpha-ESC 40A LV. We evaluated the usage of the motors MN4014 KV400 and MN4006 KV380. The performance has been evaluated using the automated test bench and an internal protocol to make the tests repeatable. In particular, the aim is

to obtain a set of characteristic curves that correlates the thrust generated with the PWM in input at various voltage levels. The protocol is the following:

- Voltage selection ranging from 16V to 25V at the step of 1V. The value is to be intended with no load;
- Execution of a PWM ramp ranging from 1150 to 1950 with a step size of 50 and step time of 7s;
- Acquisition of the data;
- restart.

The process is iterated until we obtain the motor characteristic curve for every voltage level. This helps to understand how the motors will behave at certain voltages and certain PWM ranges and can help to make predictions on how the whole UAV will perform. An initial method to evaluate if the motors fit the needs is to check the performance of the hovering of the UAV. In fact, knowing the weight of the UAV and the characteristic curve of the motors makes us able to predict the PWM range of the hovering manoeuvre. Clearly, if the hovering is at more than 50% of the PWM range we selected the wrong component with respect to the weight of the machine. Defining T_{max} as the max thrust that an UAV can perform, W_d as the total weight of the UAV, we need to obtain:

$$r_d = \frac{W_d}{T_{max}} \leq 0.5 \tag{3.2}$$

Usually, drone for racing tends to minimize r_d, reaching values like 0.12 or lower. In SAR application having a low r_d is fundamental because of PWM saturation in situations in which there are strong winds or low batteries level at high altitudes. In Airborne we achieved obtained $r_d = 0.3$ allowing enough thrust to safely conclude the search mission even with low batteries and the presence of strong winds.

The motors MN4006 KV380 offered good overall performance and was supported by the Alpha-ESC 40A LV. Nonetheless, the r_d was not low enough to guarantee enough manoeuvrability of the UAV in case of low voltages and strong winds. In fact, in that situations, the machine wouldn't have enough room in PWM to perform the search pattern and resist the wind. This motivated us to test the MN4014 KV400. The ESC firmware has been modified to be compatible with this motor and required a parametric change because of the short length of the 3 phases cables that are inside the motor group cans. In fact, in order to be fitted inside the cans, the 3 phases cables connecting the motor to the ESC are shortened.

FIGURE 3.9: *The MN4014 KV400 electric moto produced by TMotor*

This modified the impedance between ESC and motor and has to be taken into account in modifying the parameters of the ESC firmware. The extraction of the motor characteristic curves (in fig. 3.12) and surface (in fig.3.13) allowed us to directly correlate the performance of the MN4014 to the ones of the MN4006 by imposing the data of the thrust profile obtained with the MN4006, in order to check the corresponding values in PWM and voltages of the MN4014. This method is useful to directly measure the difference in performance between two motors.
The MN4014 KV400 turned out to be appropriate to be used as a motor in the Airborne's UAV. The promising performances have been validated during real flight tests. The current request of the motor, even if higher with respect to the MN4006 KV380, is still compatible with the used cell US18650VTC.

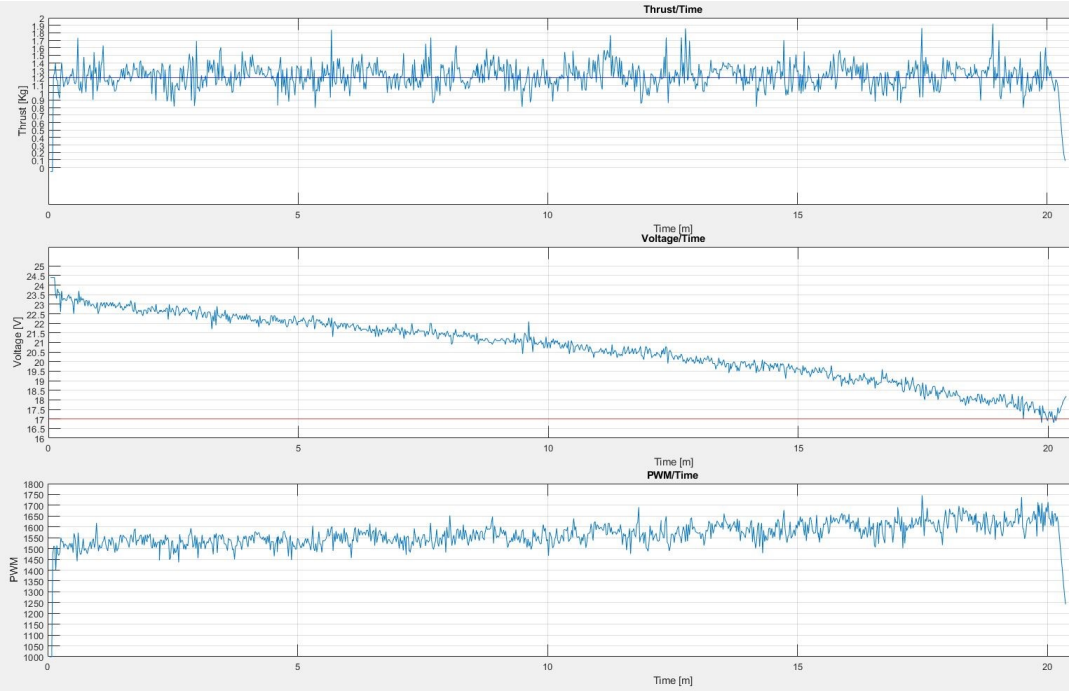


FIGURE 3.10: *Example of a discharge test. The thrust profile is extracted by a real hovering manoeuvre. The tests aim to check the discharge ratio of the battery and the increment of PWM. In fact, for a fixed thrust, with lower voltages the current and PWM increase. In extreme cases could lead to PWM saturation making it impossible to follow the thrust profile anymore and leading, in the case of a real UAV, to difficulties to manoeuvre the machine.*

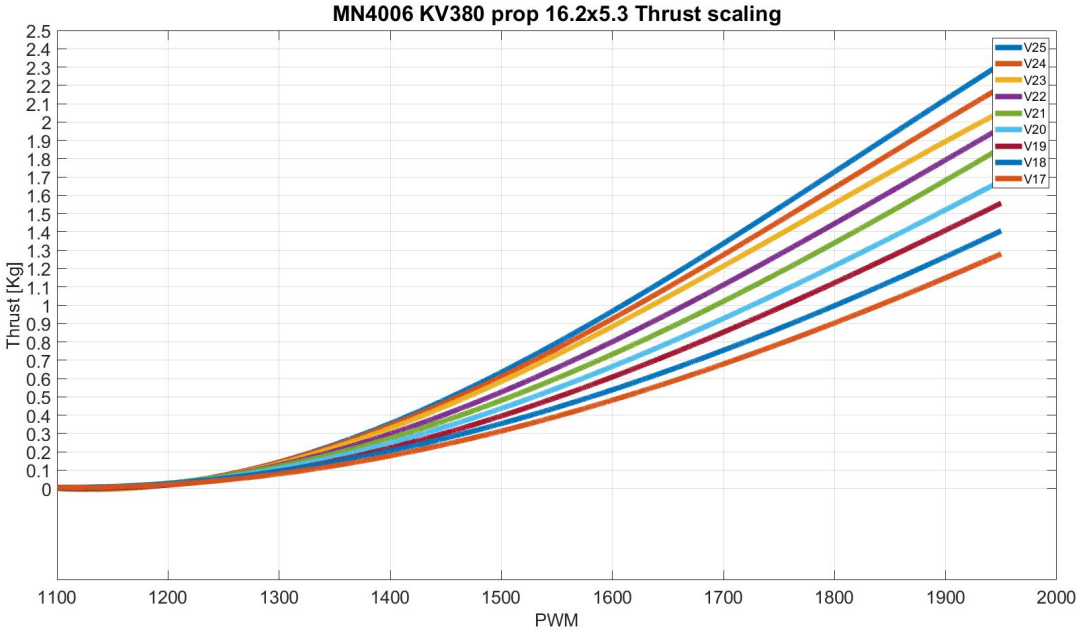


FIGURE 3.11: *In the figure, the characteristic curves extracted from motor MN4006 KV380 at different voltages. At low voltages, the motor cannot generate enough thrust to perform manoeuvres other than the landing. The test is performed using 16.2x5.3 propellers that are the most suited ones to be used with this motor. Using 17-inch props can help to improve the thrust but generate also a higher current request and less responsiveness of the machine.*

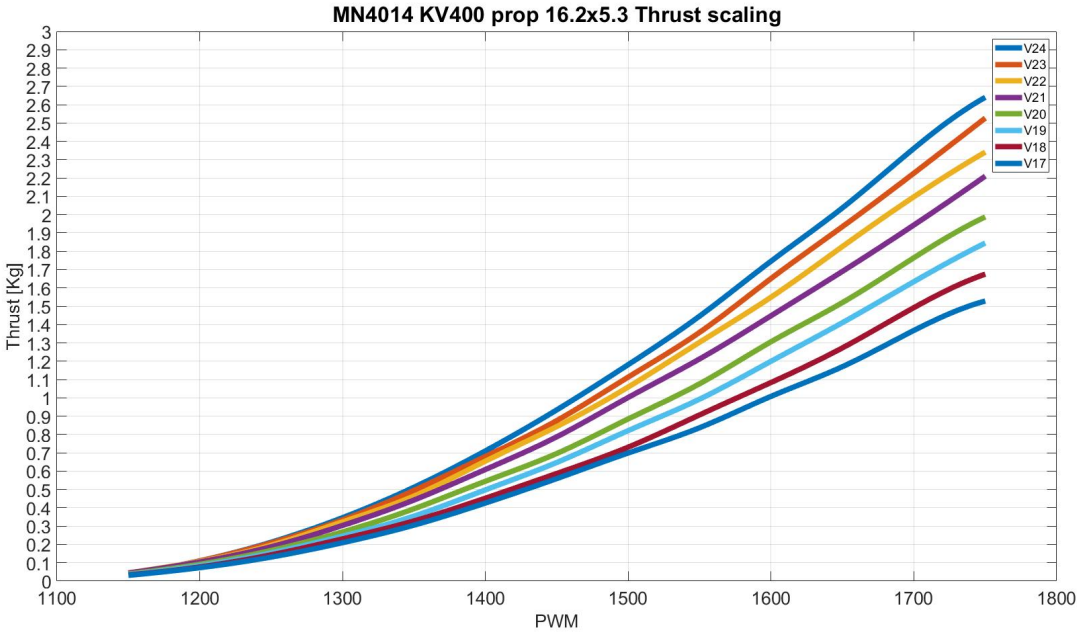


FIGURE 3.12: *The characteristic curves, extracted from motor MN4014 KV400 at different voltages. At low voltages, the motor still has a good amount of room to perform maneuvres safely, even during windy meteorological conditions. In order to highlight the difference with respect to the MN4016 KV380, the PWM has been stopped at 1750, where the MN1014 KV400 already generates more thrust with respect to the MN4016 KV380.*

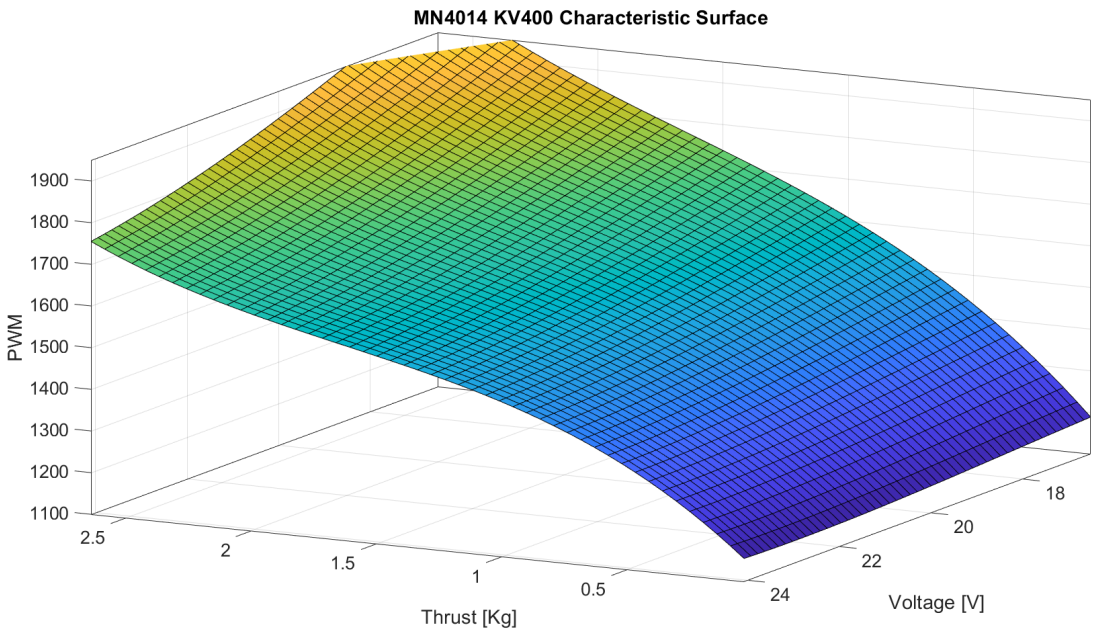


FIGURE 3.13: *The characteristic surface, extracted from MN4014 KV400. The discrete data are interpolated using SPlines. The data of the thrust, voltage, and PWM can be used to generate a surface that explains the behaviour of the motor. Given a certain PWM profile is possible to obtain the corresponding thrust profile at certain voltage levels. This makes it possible to predict the behaviour of the motor group and gives an important tool of comparison between two different motors.*

Propellers

The choice of the propellers can change completely the performance of a motor group. Pitch and diameter are two key factors in choosing the right pitch for the application. The pitch is the angle between the blade of the propeller and the perpendicular to the motor axis. It is possible to picture the pitch as the travelling distance of a single revolution of the propeller if there are no losses. A high-pitched propeller will move more air and generate more thrust at the cost of creating more turbulence. It follows that they are suitable for applications in which the UAV has to fly at high speeds. Instead, low-pitched propellers will generate more torques and more stable and efficient flights at low speeds. The diameter is related to the air contact. Increasing the diameter of the propeller will lead to less responsive bust stable flights. Instead, a low-diameter propeller will lead to more responsive flight, since they require less effort to be speeded up or slowed down. Besides, when selecting propellers attention must be put into the power consumption of the motor. In fact, given a fixed value of throttle different propellers will produce different power consumption, turns, and thrust.

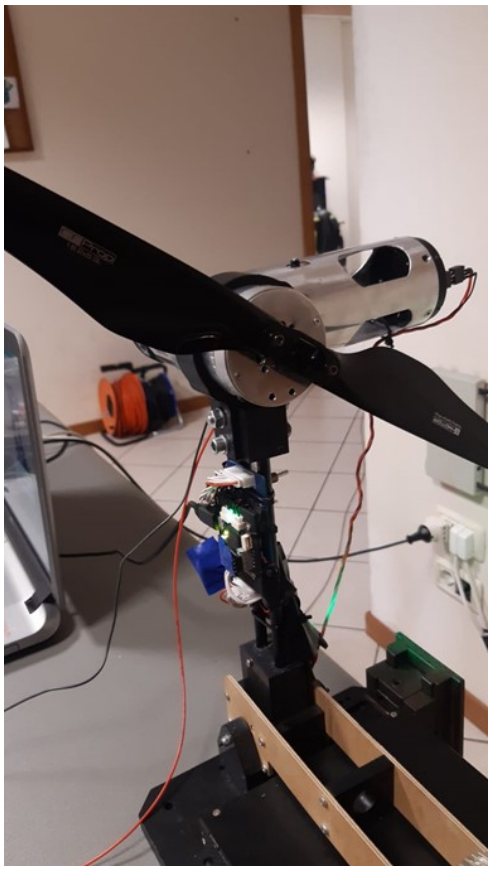
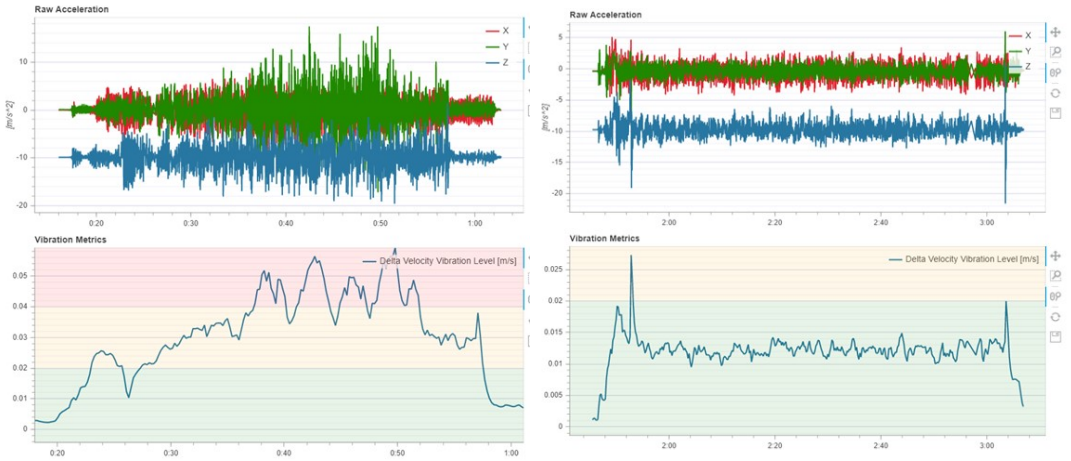


FIGURE 3.14: *The test bench used to check the vibration caused to the autopilot by a non perfectly balanced propeller.*

During the Airborne project, the encumbrance of the machine was a key factor. So, we decided to use foldable props. In particular, considering the application, we selected the 16.2x5.3 foldable propeller. The flight test with the propellers presented high vibration. Sometimes a small unbalance in the propeller's blades can cause those types of vibrations. Consequently, in order to check if a unbalance in one propeller could cause the high vibration recorded, we arranged some tests using the motor test bench. The setup was composed of the test bench and one autopilot PX4, rigidly attached to the arm of the test bench, in order to collect data using the internal accelerometers. The results highlighted the fact that some blade was unbalanced and was generating a high amount of vibration to the autopilot. Attention is drawn

to the fact that an excessive amount of vibrations could cause loss of flight efficiency, degrades in-flight performance, and even loss in the position estimate. A quick fix to this problem could be the application of some amount of tape on the unbalanced blade. Nonetheless, given the delicate application in which the UAV would operate we decided to replace the unbalanced blade with new balanced ones.



(A) (B)

FIGURE 3.15: *The results of some tests to solve the high vibration problem. In* (A) *the test with unbalanced propellers. In* (B) *the result of the test with new balanced propellers and improved flight controller's dumper.*

3.2.2 EMI Tests

The usage of the ARVA antenna as payloads imposed the execution of additive tests when selecting the power train's components. In fact, despite the shielding procedure described in ch. 2.4 the test bench has been used to individually test the EMI on the ARVA antenna of the single power train components. The usage of the test bench to check the EMI on the antenna allowed to save time and to obtain repeatable results. In particular, any modification to the frame structure has been tested as well as the implementation of new components or the modification of some part of the electronics. The tests have been executed following the same protocol defined in cp. 2.4 but the firmware to acquire the ARVA data has been modified. In particular, part of the functions of the test bench firmware in cp. 3.1.2 has been integrated with the ARVA acquisition shell. The data acquired are, then plotted and analyzed through a Matlab script.

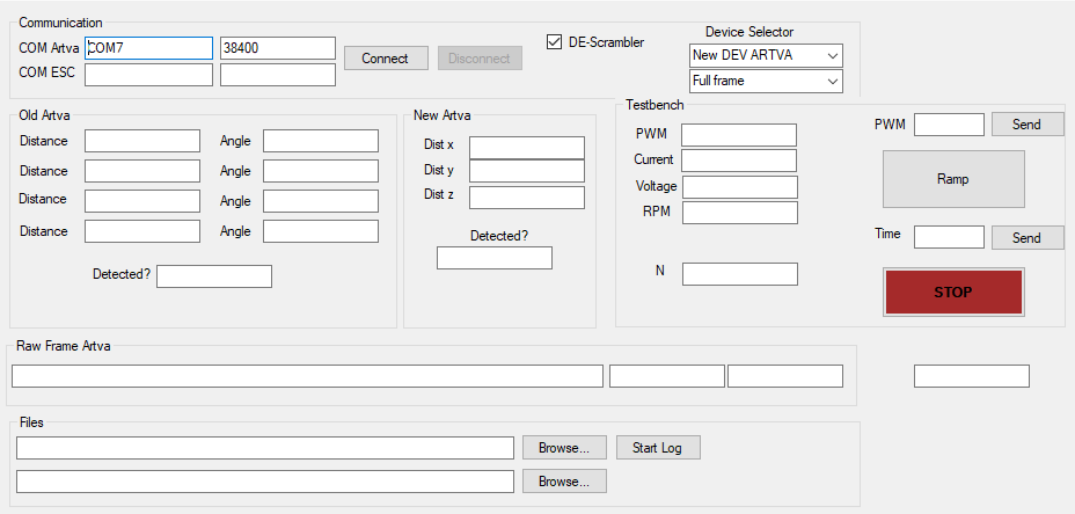


FIGURE 3.16: *Example of the ARVA data acquisition shell. The shell has been integrated with some functionality of the test bench's shell to be able to command the motors when recording the ARVA data.*

The setup obtained integrating the ARVA antenna into the test bench was useful to study the EMI of the two different motors MN4006 KV380 and MN4014 KV400. Moreover, it was of fundamental importance in determining how positioning the antenna respect the drone's frame structure. In particular, different tests has been made, using the same PWM profile and the same voltage, with different position of the ARVA with respect to the motor in order to understand at which distance the readings of the ARVA could be considered acceptable. The results suggested placing the ARVA at around 40cm to have a substantial reduction in the EMI and were fundamental in determining the dimension of the module integration described in section 2.4.1.

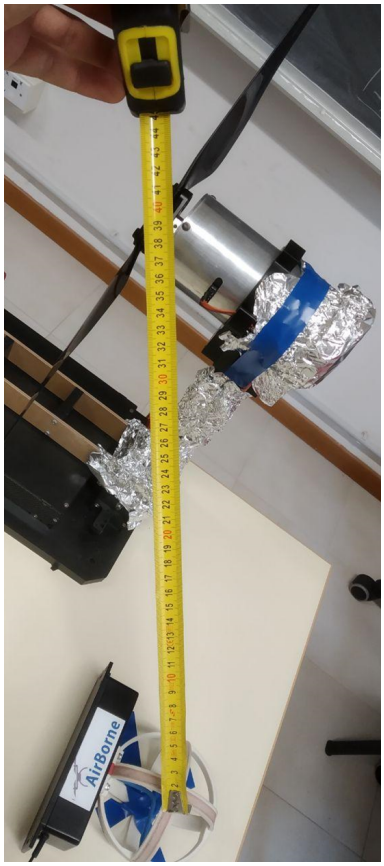


FIGURE 3.17: *The test bench used to check the distance at which the EMI is less present in the ARVA readings.*

3.2.3 Temperature Tests

The electric motors used need constantly to be cooled to avoid overheating due to the joule effect caused by the currents on the windings. The can shield in aluminium alloy does not help in this process because it does not allow the air circulation needed for cooling the motors. If stressed enough, the motors in this condition could reach high temperatures that may cause damage to the motor itself and its surroundings. In fact, prolonged exposure to high temperatures can demagnetize the magnets of the motor, degrade the structure of the motor group, and damage all the components inside the motor group. In Airborne we faced this problem when, during the flight carried out on 25/11/2019, by the CAI, one motor of the drone broke the frame and moved away from its axis of rotation, causing the drone to fall to the ground. The circle of PVC that holds the motor was deformed. The main suspect was the heat that might weaken the structure of the circle.

In order to resolve the problem, we arranged a study using the test bench, a thermal camera, and some temperature sensors. The test has been done using both an external power supply and batteries. The test aimed to reproduce the exact condition of the flight that led to the crash. In doing so we extracted the PWM profile of the flight and used a fully charged battery. The final temperature reached 90°C at the end of the test. Since the PVC circle was designed to work under 80° we decided to replace it with an aluminium plate that, since is thermally connected with a thermal paste to the external frame, might also act as a heatsink.

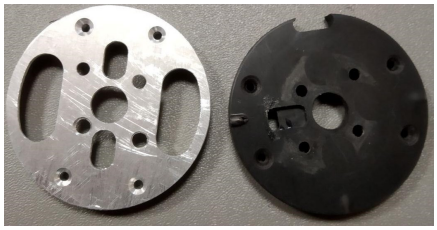
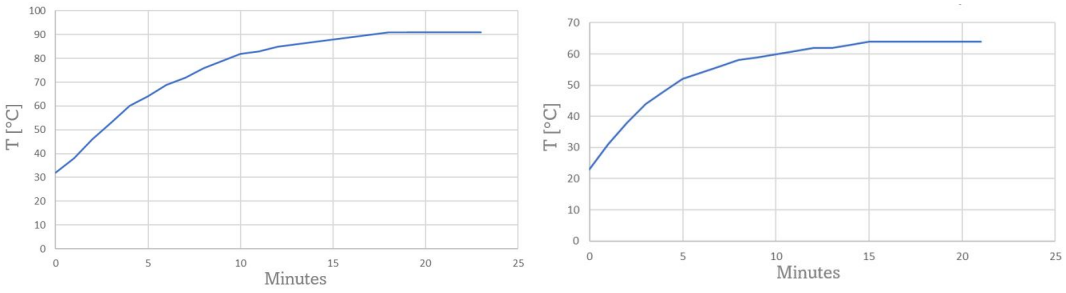


FIGURE 3.18: *The plates that sustain the motor inside the motor group can. In the left part, there is the aluminium plate designed to avoid overheating problems. On the right, there is the PVC plate which can be deformed by excessive heat.*



(A) (B)

FIGURE 3.19: *The results of the temperature test. In* (A) *Using the PVC plate the temperature reaches high and dangerous values. In* (B) *the results of the test using the aluminium plate and the thermal paste. The temperature reaches a steady state at 60°C, which is an acceptable value.*

The usage of thermal paste, in order to thermally connect the components of the can, was suggested from the fact that using the thermal camera, we noticed a poor heat spreading among the can structure. The results of the new implementation, in fig. 3.19b, was good enough to be implemented in the Airborne's quadrotor, in particular on the AlCan01. In fact, with the development of the AlCan02, we decided to slightly modify the can structure and the position of the motor, in order to eliminate the need for the aluminium plates and still contain the temperatures. We developed the new motor can with some holes in the top part and on the sides to favourite the air replacement to cool down the motor. Thanks to the application of the test bench to study the EMI acting on the ARVA in cp. 3.2.2 we were able to check that the modification was not degrading the shielding performance. Additionally, we decided to mount the motor upside down in order to install it directly on the top plate of the can structure. The benefits are twofold: there is no need for aluminium plates, this makes the drone more compact. Besides, being the top plate directly in contact with the air of the propellers allows a better heatsink action. The modification allows a steady-state temperature of 50°C in tests at room temperatures. Is worth noticing that, since the UAV will operate in alpine environments where there are subzero temperatures, the steady-state temperature should be even lower.

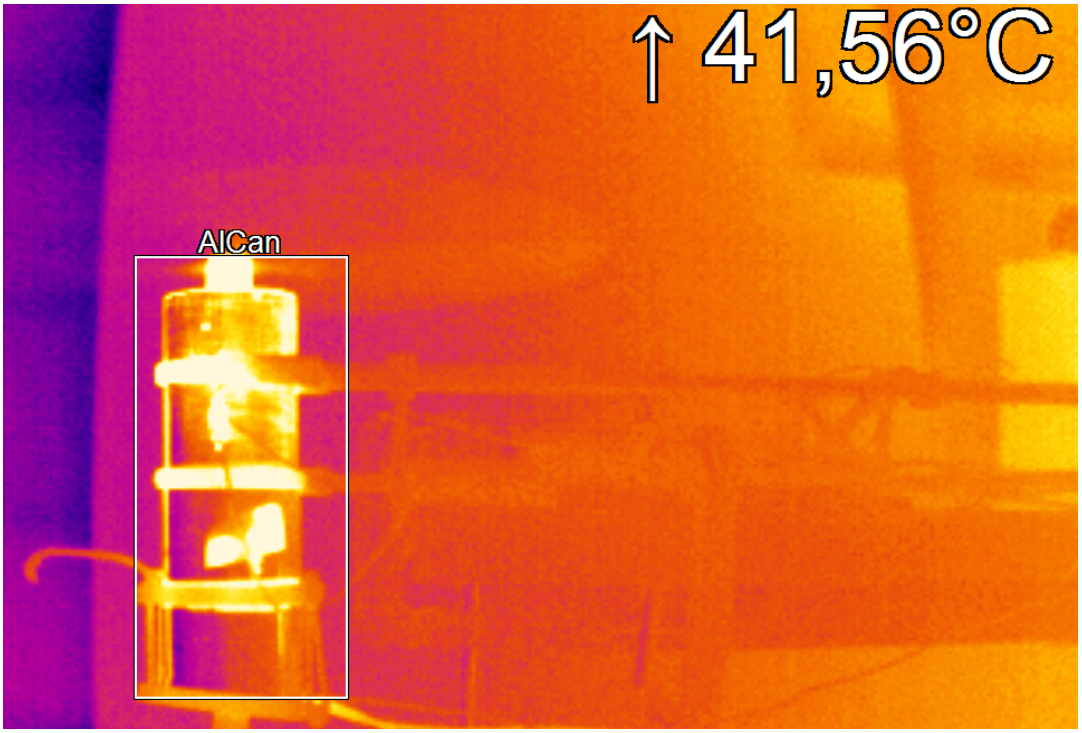


FIGURE 3.20: *The thermal camera measurement during a temperature test with the PVC plate in AlCan01. Due to the poor thermic exchange the external temperature measure 41.56°C while the temperature sensor inside the can measure more than 80°C.*

3.3 Power Train Structure

Given all the components selected with the previously described procedures, and given all the modifications to the motor group can structure, the power train can be assembled. The structures are enumerated based on the motor numbers, following the convention described in Appendix A.
The final power train structure is composed of the following components:

- Aluminum can structure with holes for air replacement;
- 6-S Battery pack with Sony US18650VTC cells ;
- internal PCB;
- Arduino Pro Mini for data handling;
- ESC Alpha 40A LV;
- DC Motor MN4014 KV400;
- 16.2x5.3 foldable propeller.

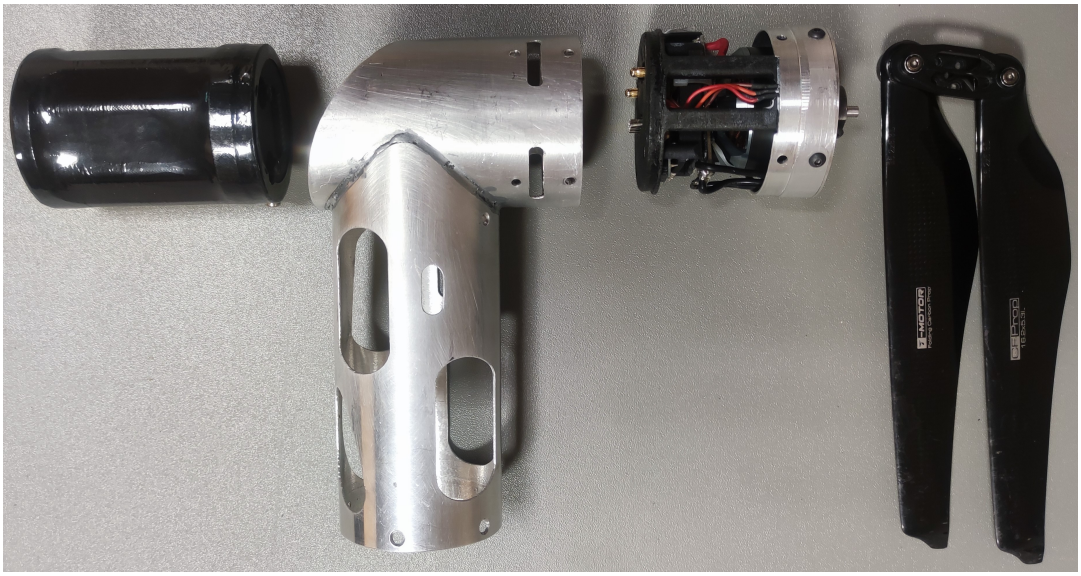


FIGURE 3.21: *The structure of the power train module. In the image are present: the battery pack, the aluminium frame structure, the motor group, and the foldable propeller. The motor group contains the PCB, ESC, and the motor. Inside the frame structure, there is a connector that needs to attach the structure to the central frame of the UAV.*

3.4 Avionics

As well as for the motors, the avionics has been tested and chosen to fit the particular needs of the Airborne's UAV. In detail, the avionics has to be compatible with the modular shape of the UAV and needs to have the following features:

- Presence of a mainboard for data processing;

- Presence of a deployment system;
- Presence of an autopilot connected to the mainboard;
- Presence of sensors for terrain-following;
- Presence of a system for data storing;
- Presence of a radio module;
- Integration with payloads;
- Integration with PCBs.

In fact, in order to be used in a SAR operation, the UAV needs to have a system that handles all the telemetry data, coming from the motors and from the sensors and sends them to the RC for real-time supervision. Additionally, the data must be saved in order to be analyzed offline, if needed. This implies the introduction of a board that collects all the data and processes them, the presence of an RC for remotely controlling the drone and visualizing the main information, and a data storage system. Furthermore, the UAV must perform automatic paths and must have safe related flight routines (i.e. RTL) and automatic flight modes. So, there is the need to have an Autopilot. Since the UAV will be operated on avalanches sites, the barometric altitude estimate and the GPS altitude estimate are not enough to perform a terrain-following over highly irregularly sloped terrain, so there is the need to equip terrain-following sensors. All the avionics has been divided into two major groups: The CanSX group and the CanDX group. Each of the two groups has unique features and fits different needs.

3.4.1 CanSX

The group contains the mainboard used for data processing and payload handling, the radio module, the data storing system, and the deployed system. All the group components are integrated into a PCB designed by Aslatech. Besides containing the various modules, the PCB is also connected to motor group 2 and to motor group 3. This allows to exchange data with the motor groups through a serial port and allows to power up the PCB's components through the 24V line coming from the batteries. Through DC/DC, the 24V line is converted into a 5V line, usable by the boards installed onto the PCB. In order to prevent, in case of an imbalance in the motor groups' batteries levels, that the current flows from the most charged battery into the less charged battery, we installed two diodes in the PCB. In the AlCan01 we used the first version of the PCB without the diodes, in order to obtain a self-balancing circuit. Although having a self-balancing circuit could be beneficial and prevents crashes in case of losing one battery, we decided to eliminate the feature to preserve the health of the electronics components. In fact, having the load current flowing directly into the PCB can damage the attached electronic components other than cause more EMI on the ARVA payload. The mainboard is a Mega2560 PRO-MINI Module 5V ATmega2560-16AU, installed next to an Arduino Pro mini to handle the Mavlink

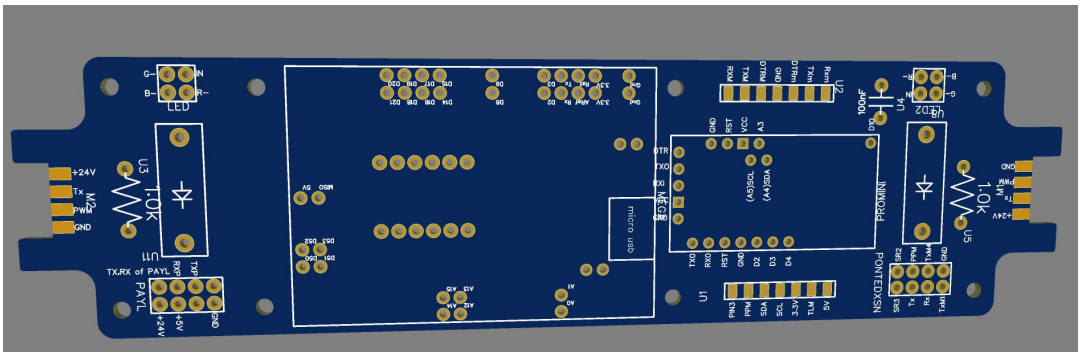
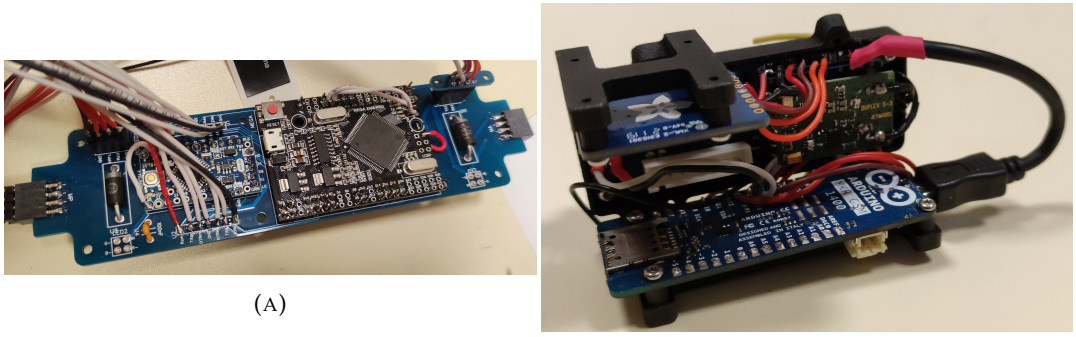


FIGURE 3.22: *3D model of the CANSX's PCB. All the avionics is installed into this PCB in order to avoid having a complicated cable system that can cause interference problems to the RECCO antenna and can be difficult to handle and debug in case of faults.*

conversion of the messages exchanged with the radio module. Two programmable RGB LEDs are attached to the 5V line. The LEDs serves to identify the quadrotor's sides and are mandatory according to the last ENAC rules.

Connected to the PCB there is a second PCB containing the radio module, the payload module, and the data storing module. The radio module is composed of a Jeti antenna paired with a JETI Model DS-12 RC transmitter. The data storing module is composed of an MKR 1400GSM with attached an SD module. The function of the MKR is threefold: it handles the data logging into the internal SD card, communicates the data on a server through GSM connection, signals the GPS position of the quadrotor to the cellphones of the rescue team. The deployment system was a combination of two servos that, based on an RC command, could deploy markers to signal the position of the buried victim, localized using the payloads. According to the new ENAC rules, is not possible anymore to eject objects from the quadrotor. So, we use the MKR 1400 GSM as a deployment system. In fact, by hovering on the position of the buried victim and sending the GPS position through GSM to the rescue team is possible to signal the position of the victim without the ejection of objects from the UAV. The payloads, mechanically integrated into the central CAN of the UAV, are directly connected to the mainboard.



(A)

(B)

FIGURE 3.23: *The components of the CanSX. In* (A) *The main PCB with the Mega2560, the Pro Mini, LEDs, DC/DC, and diodes. Besides, all the connections that are needed to interface with the other modules. In* (B) *the PCB contains the radio module, the storage module, and the deploy module. (This is the final version with no physical deployment, compatible with the last ENAC rules)*

3.4.2 CanDX

The group contains the autopilot PCB, sensors for navigation and terrain following, a board for handling the sensors data, and the GPS. Besides containing the various

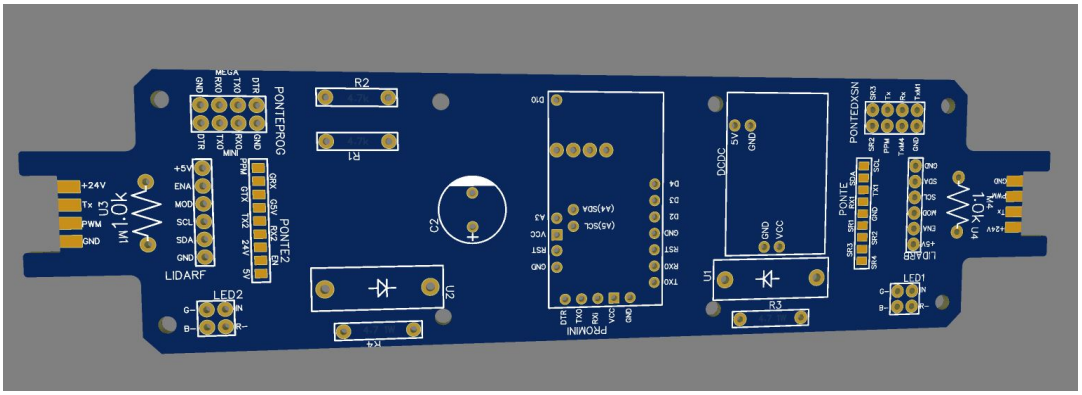


FIGURE 3.24: *3D model of the CanDX's PCB. All the avionics is installed into this PCB in order to avoid having a complicated cable system that can cause interference problems to the RECCO antenna and can be difficult to handle and debug in case of faults.*

modules, the PCB is also connected to motor group 1 and to motor group 2. As for the CanSX, this allows to exchange data with the motor groups through a serial port and allows to power up the PCB's components through the 24V line coming from the batteries. Through DC/DC, the 24V line is converted into a 5V line, usable by the boards installed onto the PCB and the autopilot. To prevent, in case of an imbalance in the motor groups' batteries levels, that the current flows from the most charged battery into the less charged battery, we installed two diodes onto the PCB. A second couple of LEDs are installed with different colours with respect to the ones installed on the left side. Connected to the PCB there is the autopilot PCB. It is composed of

a second PCB with installed one distance sensor, for terrain following, and the autopilot. The autopilot is a PixRacer. It runs all the control, guidance and navigation algorithms and provides an interface with low-level sensors. The computational unit is an ARM Cortex® M4 processor. The PixRacer autopilot includes redundant IMU for navigation. In particular it includes an Invensense® ICM-20608 Accel / Gyro (4 kHz) and MPU9250 Accel / Gyro / Mag (4 kHz). Moreover, it has a HMC5983 magnetometer with temperature compensation and a MS5611 barometer. The distance sensor is the Lidar lite, which is an optical sensor that provides ground height estimation to the autopilot and is used to implement the terrain-following feature.

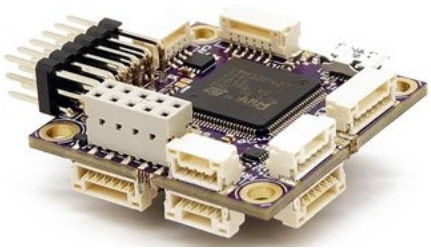


FIGURE 3.25: *The autopilot used in the Airborne's UAV.*

A three lidar configuration can be integrated inside the CanDX in order to enhance the terrain following the performance. The integration with the lidars imposes the installation of, at least, a 680 μF capacitor between the 5V line and the ground. The GPS is a high-performance external u-Blox Neo-M8N integrated by mRobotics connected directly to the PCB and then routed to the autopilot. The CanDX communicates with the CanSX thanks to a connection where are routed the PWM signals, the mavlink communication between autopilot and central Mega2560, the serial protocol containing telemetry data of the motors, the PPM signals coming from the RC.

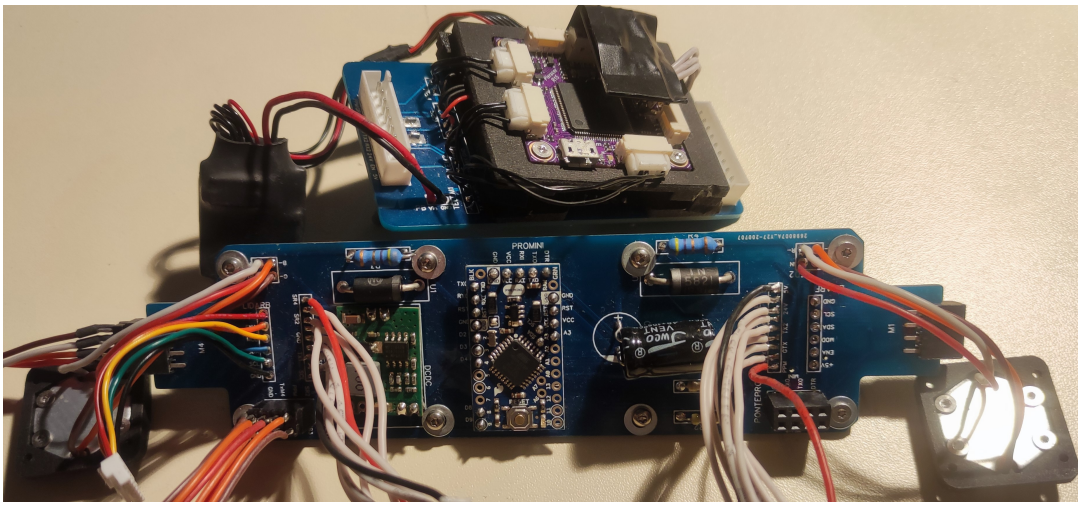


FIGURE 3.26: *The complete CanDX PCB. In the low part of the image, there is the main PCB with the Pro Mini to handle the distance sensor's data, the DC/DC, LEDs, GPS, and connectors. In the up part of the image, there is the autopilot PCB with the autopilot and the distance sensor.*

4 Data Handling

The data exchanged by avionics and power trains groups is of critical importance in the SAR application. In fact, is needed to control the motors, supervision the performance of the power train groups, integrate and use the payloads, store the data, and so on. Basically, is mandatory to use the whole UAV.
Different protocols and data handling algorithm has been used to handle and use the data, based on the avionics and power train structures described in Chapter 3. In this chapter, we present the main innovation and implementations in handling and using the data exchanged, onboard and offboard, of the AlCan02 quadrotor. Given the nature of the Airborne project, emphasis will be given to the recognition of the payloads signals.

4.1 Drone Telemetry

The drone telemetry represents all the data that are useful to be checked, monitored or stored. The drone telemetry includes different kinds of signals ranging from the voltage level of the batteries to the flight mode of the flight controller. In Airborne we implemented custom firmware in order to check the data directly on the RC display, without the need for an additional CGS. This was a requirement of the project, because during operation in alpine environments, especially SAR operations, is not comfortable nor efficient to check data on an external device while flying and performing searching manoeuvres. The RC user interface has been divided into three main pages. On the first one is possible to check the total percentage of battery level, the distance from home, the number of satellites of the GPS, and the height above the ground. Besides, is possible to check the flight mode currently used. Additionally, on this page is possible to check all the payloads data including the payload type and the signal detected. A sound alarm is triggered, based on the value of some key parameters, in order to help the rescues to react.

FIGURE 4.1: *An example of the first page of the user interface in the RC. The user can visualize all the major data without the need to consult an external device like a CGS.*

4.1.1 Motors Telemetry

Monitoring the state of the motors and batteries is of paramount importance when flying with a UAV, especially in SAR operations. In this work, we'll refer to the motor telemetry as the data package composed by: the PWM in input to the motor, the RPM of the motor, the load current of the motor, the temperature of the motor, the voltage of the batteries, and the temperature of the battery. The motor telemetry is handled by the central Mega2560 inside the CanSX. Although being part of the motor telemetry package, the PWM inputs are generated by the Mega2560 and not collected likes the other part of the data package. The PWM inputs from motors 2 and 3 pass through the PCB of the CanSX to the ESC of the respective motor groups. Similarly, the PWM input of motors 3 and 4 passes through the PCB of the CanSX to the PCB of the CanDX where arrive in the ESC of the appropriate motors. Conversely, the other data of the motors telemetry are connected from the motor groups to the Mega2560 through a serial connection. The protocol is generated by the Arduino Pro Mini of the motor groups that, following the protocol in table 3.1, collects all the data from the ESC, merges the temperatures data and the batteries data, and communicates them to the Mega2560. The motors telemetry is then sent to the RC and to the MKR board, where is logged. All the information concerning the position of the UAV and the GPS are sent through a serial protocol from the autopilot on the CanDX to the Mega2560 on the CanSX. The payload data comes from the integration port on the central module of the UAV, and are sent to the Mega2560 through a serial protocol. The Mega2560 processes all the data received and sent e new data package to the Arduino Pro mini that modifies the protocol of the package in order to be sent

to the radio transmitter and, then, read by the RC.
The second page of the user interface is dedicated to motors' telemetry. On this page is possible to check the temperatures, voltages and currents of the 4 power train groups.

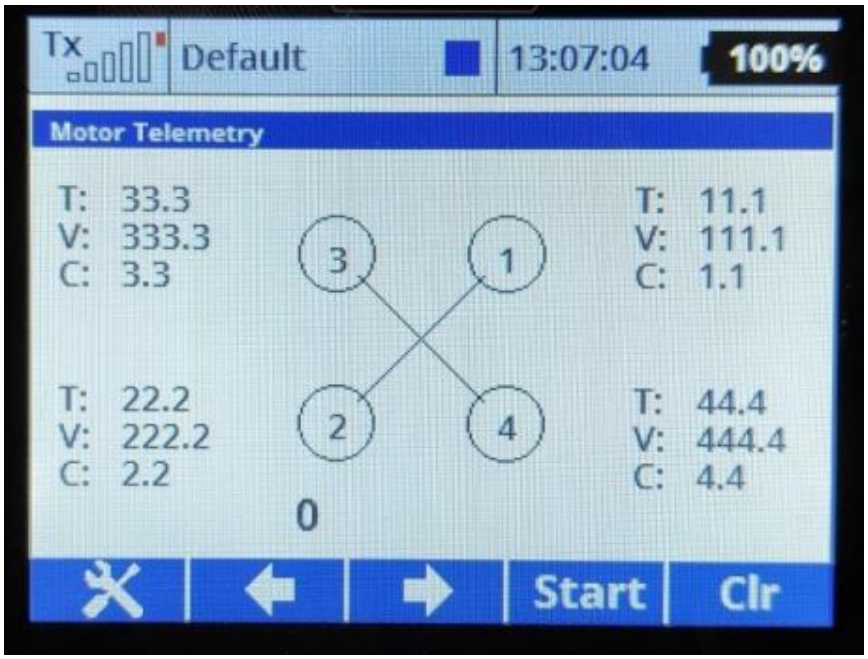


FIGURE 4.2: *An example of the second page of the user interface in the RC. The user can visualize all the major data about the power train groups.*

4.1.2 Fault System

The firmware on the RC makes use of a fault system to warn the pilot about the presence of some errors, anomalies, or important signals. The fault system can be divided into sound alarms and coded faults. The sound alarms are different based on the different triggers and are divided into three categories: temperature alarms, voltage alarms, payload alarms. The temperature alarms trigger when the temperature is over a certain safe threshold. This prevents the pilots to keep manoeuvring the UAV even in case of overheating. There are two triggers that trigger two different sound effects. The higher is the temperature the more pitched is the sound. Similarly, an alarm is triggered when the voltage percentage of the machine is under a certain threshold. They are 3 thresholds, the first one needs to signal to the pilot that there is low gas left to continue the flight, the second signals the RTL function of the autopilot, the third triggers the auto-land function of the autopilot. The RTL and auto-land are safety measures necessary to avoid crashes due to low batteries levels. As a rule of thumb, the pilot must not rely on the safety functions but must pay attention to the first audio signal and start autonomously the return to the home location of the UAV. The payloads audio alarms make easier the search operation. In fact, the pilot doesn't have to continuously watch the display in search of detection.

Instead, it can focus the attention on the UAV. The audio is different pitched based on the signal detected, as soon as a signal is detected a double beep alarm rings every second. The pitch of the sound is higher the closer the UAV is to the transmitter.

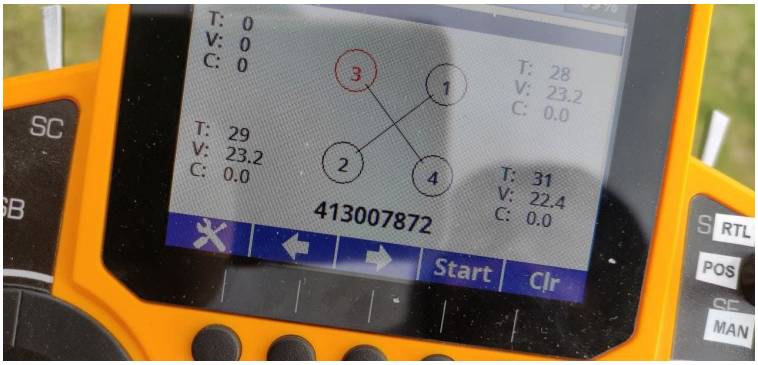


FIGURE 4.3: *An example of a fault on motor 3. The telemetry of the power group 3 is lost and does not figure anymore on the page. Detailed information about the fault can be retrieved from the fault code.*

The coded faults can be consulted on the second page of the user interface. The idea is that the Mega2560, within the data package, sends continuously a message with a fault vector. The fault vector is composed of a series of flags that represents different kinds of errors. Every flag is triggered if some condition is met. Generally speaking, the conditions are the expires of a timer or the overcoming of some threshold value. For the sake of clarity, a too high current will trigger a current fault and will turn on the flag into the vector. In the real implementation, we use a binary vector as a vector of flags. Every position represents a specific fault and the binary numbers act as flags. The vector is then expressed in decimal form into an integer number, order to be displayed on the RC. In this way is possible to communicate a large list of faults without requiring a big portion of the bandwidth available of the RC, but just sending one integer value. In fact, by having the integer error code number, converting it into a binary number, and consulting the position of flags actives using the fault list as a lookup table, we can obtain in formations about the fault sources.
As an example, let's consider the fault code on fig. 4.3 $f_c = 413007872$. By converting into a binary we obtain $f_C^b = 11000100111100000000000000000$. In order to check the sources of the faults let's look up the fault list in table 4.1. We obtain that the faults were caused by a telemetry error in motor 3. In this particular case, this was already known since missing telemetry causes also a visual fault, as can be seen in 4.3. Additionally, the fault code communicates that there is also an error in the EEPROM of the power train group 4 and, more importantly, a great current pike for all 4 motors.
The information obtained from the fault system is essential in finding possible problems before they can cause some damage or debugging what happened after a crash. This is true especially in an application like the Airborne project, where the UAV is entirely built from scratch and improved to TRL 8.

Fault list	
Fault	Vector position
FAULT_TEMP_BATT_MAX_1	0
FAULT_TEMP_BATT_MAX_2	1
FAULT_TEMP_BATT_MAX_3	2
FAULT_TEMP_BATT_MAX_4	3
FAULT_TEMP_BATT_MIN_1	4
FAULT_TEMP_BATT_MIN_2	5
FAULT_TEMP_BATT_MIN_3	6
FAULT_TEMP_BATT_MIN_4	7
FAULT_TEMP_MOT_MAX_1	8
FAULT_TEMP_MOT_MAX_2	9
FAULT_TEMP_MOT_MAX_3	10
FAULT_TEMP_MOT_MAX_4	11
FAULT_VOLT_MIN_1	12
FAULT_VOLT_MIN_2	13
FAULT_VOLT_MIN_3	14
FAULT_VOLT_MIN_4	15
FAULT_VOLT_DIFF	16
FAULT_CURR_MAX_1	17
FAULT_CURR_MAX_2	18°
FAULT_CURR_MAX_3	19
FAULT_CURR_MAX_4	20
FAULT_TELEM_1	21
FAULT_TELEM_2	22
FAULT_TELEM_3	23
FAULT_TELEM_4	24
FAULT_EEPROM_1	25
FAULT_EEPROM_2	26
FAULT_EEPROM_3	27
FAULT_EEPROM_4	28

TABLE 4.1: Fault list

4.2 Signal Recognition

Identifying the signals of the payloads is essential in achieving the integration of the payload. The UAV must analyze the data received while performing the search patterns. In case of a presence of a transmitter (for the ARVA) or a reflector (for the RECCO), the UAV must communicate the detection to the pilot along with the information about the data. The data is communicated to the UAV and then to the pilot depending on the particular payload.

4.2.1 RECCO Recognition

The RECCO module analyzes the signal through an internal logic and provides a vector of data containing a timestamp, signal to noise ratio of the strongest pulse,

signal strength, background noise of the antenna. Additionally, like the commercial RECCO antenna, it provides audio output. In the RECCO integration in the UAV, we take the outputs signals of the antenna and process them in the Mega 2560 inside the AlCanSx, as described in 3.4.1. Since the antenna already provides information about the signal detected we use just pass this information to the pilot through the RC.

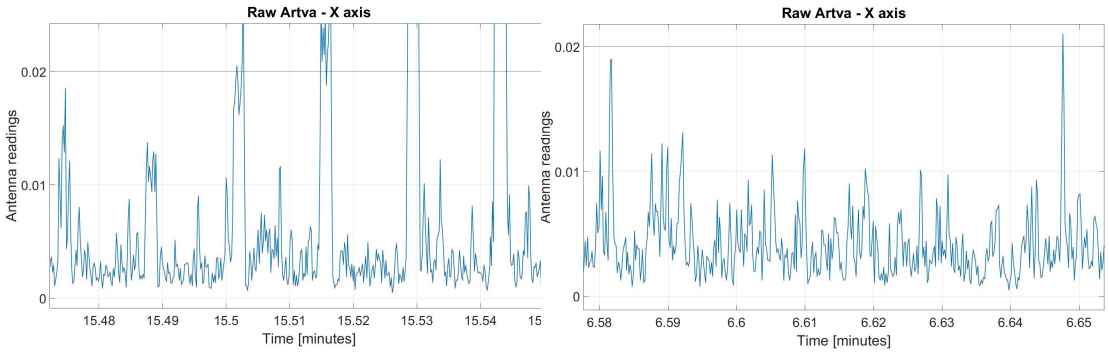
FIGURE 4.4: *An example of detection of the RECCO signal during internal tests. The RECCO emissions gain can be modulated through a channel of the RC. In the example is settled to 4.*

In particular, the power of the signal detected is displayed on the display of the RC with the gain of the antenna. Additionally, in order to recreate the feeling of the commercial RECCO device we reproduce the audio signal on the RC. The audio signal is sent to the RC where, through the firmware, we can reproduce a similar beep that is heard when using the traditional RECCO antenna. This is a useful feature because the pilots might already be trained to use the RECCO and so are used to perform search operations with it. Besides, having an audio signal allows the pilot to focus on the flight instead to watch always the display.

4.2.2 ARVA Recognition

The ARVA module processes the signal captured by the 3D antenna by an internal analogical filter and provides a vector containing three power-related signals of the

three-axis of the antenna. The output signal is then transformed into an optic signal, though a transducer, in order to not generate EMI acting on the antenna. The optic signal is then re-transformed into an electric signal that, through the payload interface of the UAV, is processed by the Mega 2560. The data of the three-axis has to be processed in order to determine the presence of the beacon and eventually its position. The presence of EMI complicates this task because they modify the reading of the various axis making indistinguishable the presence of a beacon if the interference is too high. An example can be seen in fig. 4.6 where the presence of the noise makes barely distinguishable beacons at more than 20m of distance. As explained in cp. 2.2 a 20m distance, on a certain axis line, can be associated with the power-related value 0.02. The shielding action of the UAV makes the action of the EMI low enough to make us able to recognize the beacon signal in a 20m range.



(A) (B)

FIGURE 4.5: *Examples of ARVA reading. In* (A) *the reading of the X-axis of the antenna when there is a beacon present. The presence of the beacon signals is clearly distinguishable thanks to the shielding of the UAV. In* (B) *the reading of the x-axis of the antenna when there is no beacon.*

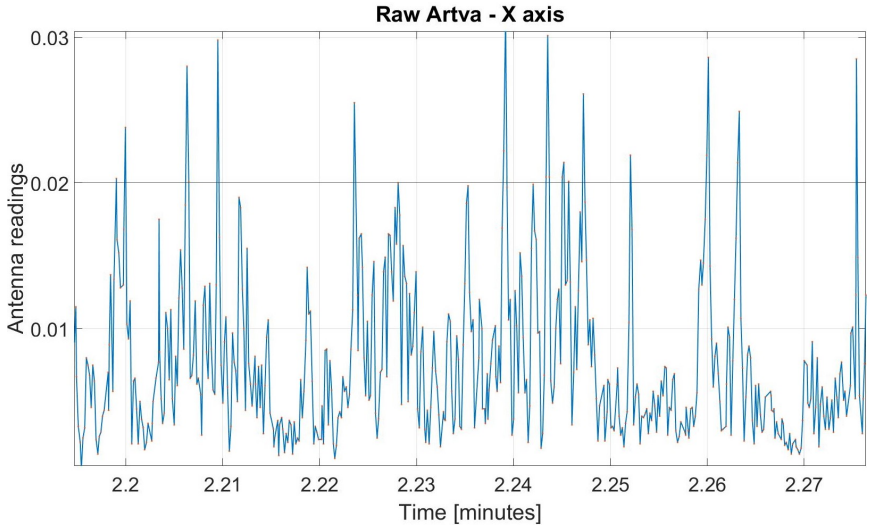


FIGURE 4.6: *An example of ARVA readings of the x-axis of the antenna when there is the presence of EMI that afflict the antenna. The interference makes it difficult to distinguish a beacon signal with an amplitude lower than 0.02.*

We developed a filtering process based on a moving filter. This approach is based on the periodicity of the beacon signals that emits pulses of 70ms duration every 1000ms with a variance of 300ms. The shape of the pulses is not a perfectly shaped signal, but as can be observed from fig. 4.5a they appear as pulses corrupted by noise. The noise is a composition of the EMI and the internal noise of the antenna. In order to evaluate the amplitude of a received pulse, we perform an averaging operation over the pulse duration. Since we use a 100Hz antenna every pulse is discretized in 7 samples. Once received a pulse with a certain duration and amplitude, we settle a window of time within which we expect to receive a second pulse. If we receive a second pulse, with a certain amplitude and duration, we signal the detection of an ARVA beacon. Besides, we imposed a ground threshold, under which every signal is considered noise and ignored. For the sake of clarity, let's define $x, y, z \in \mathbb{R}$ as the signals obtained from the antenna, p_d is the pulse duration of the beacon, w_s is the size of the window, t_d is the detection corresponding time, and s_t is the ground threshold. Then, the detection algorithm can be implemented as the following: Attention is drawn to the fact that in case of a massive presence of EMI the

Algorithm 1 Onboard ARVA detection algorithm

Require: x, y, z, p_d, w_s, t_d
Ensure: $x, y, z > 0$
  $mod = \sqrt{x^2 + y^2 + z^2}$
  $mod_d = duration(mod)$ ▷ Function to extract the pulse duration
  $mod_m = mod / mod_d$
  $w_m = 0$
  if $mod_m \geq s_t$ & $mod_d \geq p_d$ **then**
    if $t_d \leq w_m$ **then**
      if $mod_d \geq p_d$ **then**
        DETECTION
      end if
    else if $t_d > w_m$ **then**
      $w_m = t_d + w_s$
    end if
  end if

antenna will capture a higher gain noise. Consequently, some pulses generated from EMI sources could be classified as beacon pulses, leading to false-positive detection. In order to avoid the false-positive classification, we can tune the parameters of the filtering process. In particular, by increasing s_t is possible to exclude the majority of noises but at the cost of restricting the detection radius. Instead, by increasing w_s is possible to check for more correspondences between pulses, making it less likely that they are caused by external sources different from the ARVA beacon. Is worth noticing that this will come at the expense of the detection time. In fact, it would require spending more time in acquiring the signal before triggering the detection. This directly afflicts the search strategies, because a bigger detection time implies a lower speed of the quadrotor.

As an example let's suppose to have a detection time of $3s$ and a detection radius of $20m$. Additionally, suppose to have a quadrotor positioned at the centre of a reference frame where the z-axis is parallel to the motor axis and passes through the centre of mass of the quadrotor. Suppose to have a beacon placed at coordinates $[18m, 8.71m]$ with respect to the centre. Finally, suppose that the UAV is moving towards the y-axis of the reference system on a straight line with a speed of $7m/s$. A detection time of $3s$ means that the beacon has to lie inside the detection radius for $3s$ in order to be detected. This implies that after $3s$ the UAV is moved of $21m$ on the y axis, so the beacon is outside the detection radius. In this situation, due to the detection time, flight speed, and radius, the algorithm would not detect the presence of the beacon. Similar situations are taken into account during the development of search strategies. It is trivial that a smaller detection time would make it less likely to lose a detection but, in order to guarantee a detection probability of 100% when a beacon enters in the detection radius, it should be zero. As anticipated, another solution could be to reduce the speed of the quadrotor but this will come at the cost of a slower search operation. Consequently, redundant paths are usually executed, where the UAV covers a certain point multiple times making it less likely to miss detections.

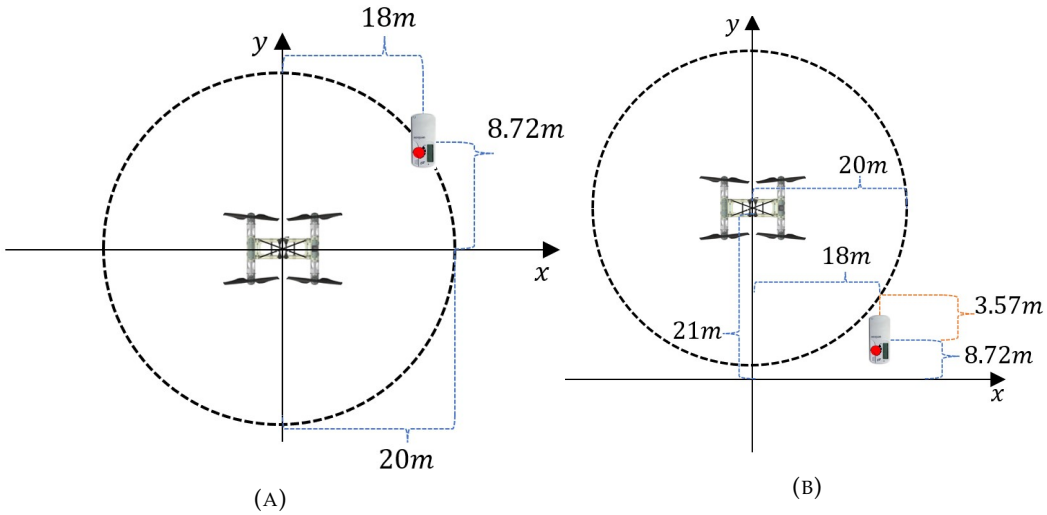


FIGURE 4.7: *Examples of ARVA missing detection. In* (A) *The start position of the example. The beacon is placed at* $[18m, 8.72m]$ *from the quadrotor. In* (B) *the results after 3 seconds of flight at a constant speed of* $3m/s$ *along the y-axis. Since the beacon is outside the detection radius it can't be anymore detected.*

In the Alcan02 we implemented $w_s = 2.6s$, a detection radius of $20m$ and search patterns with redundancy. If the algorithm triggers a detection, the amplitude of the modulus is translated into a distance. In order to do so, we linearized, inside the interval of interest (i.e from 0m to 20m), the relationship between the power-related signals of the antenna and distance from the beacon during similar tests as the one reported in fig 2.9. After that, the data is sent to the RC where is visualized in the display with graphic indications. Similarly as did for the RECCO, the RC triggers

an audio signal that is more pitched the less is the distance. This helps the pilot to keep track of the detection of the ARVA without the need to continuously watch the display of the RC.

FIGURE 4.8: *An example of detection of the ARVA signal during internal tests. The value displayed is the estimated distance from the transmitter. The firmware is already implemented for detecting multiple ARVA signals.*

4.2.3 ARVA Neural Network

The algorithm developed to detect the presence of an ARVA signal does not tell to the rescue the position of the transmitter, but only the distance value. According to the test conducted by CAI, this is sufficient data to start the rescue operation in a delimited area. In fact, usually, the rescuers start the rescue manoeuvres as soon as in a certain area there is a detection of the signal. This happens because the buried victim could be separated from the transmitting device, or there could be other victims in the area. Therefore, the area is searched even if the rescue knows the exact position of a victim. Nonetheless, is worth noticing that, the exact position of a transmitter

could be used as a starting point to search a delimited area.
Due to EMI produced by DC motors and avionics, implementing a reliable estimation algorithm of the position of the transmitter is a challenge. Nonetheless, thanks to the usage of the shielded UAV, we developed a multi-output multivariate regressive convolutional neural network in order to provide the distance between UAV and transmitter.

Time Series

The problem of analyzing the ARVA data and extracting the information of the position of the transmitter can be cast as a time series classification problem. Providing formal definitions for time series classification problems:

Definition 4.2.1. A univariate time series $X = [x_0, x_1, x_2, \ldots, x_T]$ is a ordered set of values.

Definition 4.2.2. A multivariate time series $\Delta = [\delta^1, \delta^2, \ldots, \delta^M]$ is a vector composed by M univariate timeseries $\delta^i \in \mathbb{R}^T$ with $i \in [1, M]$.

Consequently, by considering the ARVA signal's natural temporal ordering, we can define the acquisition time T and the ARVA time series vector $\Delta = [\delta^1, \delta^2, \delta^3]$, where each element $\delta^t \in \mathbb{R}^N$ is a univariate ARVA time series composed by the ARVA signals. Is worth noticing that each value of the Δ vector corresponds to a measured single-axis signal during the acquisition time T.
In an ARVA identification problem, the ARVA signals are not the only time series that we are interested in. The altitude of the UAV, and its attitude, can modify how the antenna receives the signals. Those data are available in the Airborne's UAV and can be used to extract a correlation with the antenna's data. Due to possible differences in the timestamp and logging time, there might be the need to synchronize the signals, in order to obtain time series with a common T.

Convolutional Neural Network

Among the hundreds of methods that are present for the identification of time-series signals, recently the usage of NNs is spreading. In fact, the high performance and the relatively simple data-based approach make them suitable to be applied in the data extraction of recognition tasks with time series [24]. Among these, convolutional NNs proved to be very performing, not only in feature extraction from images [31] but also from time-series signals like speech or music [3] [33] with no prior feature engineering [4]. In fact, the convolution operation can be seen as a sliding filter over the time series. In operations with images, the filter has two dimensions. Instead, when processing time-series data, the filter exhibits just one dimension. Additionally, CNN can perform classification or regression from high-dimensional raw data. In fact, different works used regression CNN to perform signal detection and localization, even with noise [14][1].

The CNN structure is composed of three main layers: convolutional layers, pooling layer, fully-connected layer. Differently from fully connected neural networks, the fully connected layers are absent in the input and hidden layers of the network but can be found as an output layer. Based on the type of outputs, the network will perform classifications or regressions. The convolutional layers take in input a matrix whose values represent images or a time series. A feature detector, named Kernel, slides on the matrix applying the convolution operation. By defining $U \in \mathbb{R}^{m_h x m_w}$ as the input matrix, $X \in \mathbb{R}^{n_h x n_w}$ as the kernel matrix, and $Y \in \mathbb{R}^{p_h x p_w}$ as the output matrix, the convolution operation is given by:

$$Y(i_y, j_y) = \sum_{i=0}^{n_h-1} \sum_{j=0}^{n_w-1} X(n_h - 1 - i, n_w - 1 - j) \cdot U(i_y + i, j_y + j) \tag{4.1}$$

where, $i_y \in [0, p_h - 1]$, $j_y \in [0, p_w - 1]$ are generic integers that indicates a generic element of the matrix Y. It is possible to express the relation between the dimension of the output matrix such that $p_h = m_h - n_h + 1$ and $p_w = m_w - n_w + 1$. It is worth stressing the fact that this operator can be also applied, as a derived case, to time series. The filtering operation offers two degrees of freedom represented by stride and paddling. The stride defines the amplitude of the horizontal and vertical shift of the Kernel during the operation of each $Y(i_y, j_y)$. The padding defines the value of the extra lines added at the boundary of the input matrix. In fact, one tricky issue when applying convolutional layers is that we tend to lose data on the perimeter of our image. Since we typically use small kernels, for any given convolution, we might only lose a few data, but this can add up as we apply many successive convolutional layers. One straightforward solution to this problem is to add extra pixels of filler around the boundary of our input image, thus increasing the effective size of the image. By introducing s_f, p_c, and p_r respectively as the value of strife, columns of padding, and rows of padding, we obtain the dimension of the output matrix in the following form:

$$size(Y) = \left[\frac{m_h - n_h + p_r + s_f}{s_f}\right] \times \left[\frac{m_w - n_w + p_c + s_f}{s_f}\right] \tag{4.2}$$

It is worth noticing that, if we select unitary strife and no padding we obtain the standard dimension of the output matrix. After the operation performed by the convolutional layer, we obtain a new matrix Y that is processed by the pooling layer. The main purpose of this layer is to gradually reduce the size of the matrix. This happens because the entire operation tends to eliminate all the redundant and not significant information of the matrix. In this way is possible to extract and propagate only the useful data. The main advantage is to gradually work with lighter matrices, making the computation faster. Besides, this allows increasing the spatial invariance to input shifting, making the network more robust. Similar to what happens in convolutional layers, a pooling window slides on the input image and

reduces the section to just one element. The value of the selected element could be the maximum between the values covered by the pooling window, or an average. The training processes of the network modify the values of the filters used in the convolution operation. Different datasets might lead to different values of filters. It can be deduced how good training could help to extract essential features from the time series. Applied in the ARVA case, this could lead to identifying the presence of the signal or correlating the ARVA time series signals with the distance from the receiver. Inspired by the huge work done in [20] collecting a time series dataset and

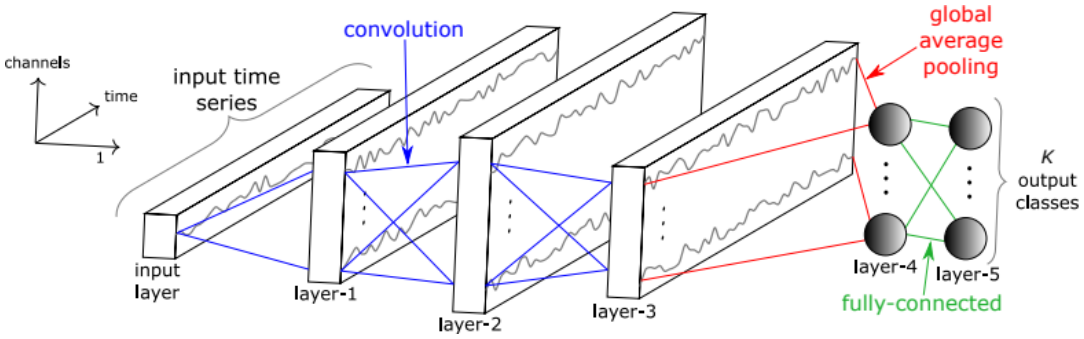


FIGURE 4.9: *An example of a CNN structure with 3 convolutions layers and 2 fully connected layers. The image is taken from* [27].

implementing a common NN creation and testing procedure, we decided to use a NN to extract the data concerning the position of the transmitter, from the ARVA signals. In particular, motivated by the good performances of CNN in feature extraction from time series [27] [1] we decided to implement a CNN. The CNN can be, additionally, defined as multivariate if we use a multivariate time series as an input, multi-output if we obtain multiple outputs from the network, and regressive if we obtain regressive output vectors from the network.

Regressive Convolutional Neural Network for ARVA Detection

We decided to implement a NN for extracting the information regarding the distance between receiver and transmitter from the ARVA signals. In particular, we aim to implement a multi-output multivariate regressive convolutional neural network. Since the training process is a supervised one, we need to collect enough data to perform an efficient training process. A lack of differentiation, or number, in data could easily lead to overfitting. We implemented the network using Tensorflow 2.6 and a computer with GPU GTX 1070 and CPU Intel I7-7700HQ @ 2.80GHz. Our setup to obtain the data to develop the RCNN was composed by:

- 100Hz 3D ARVA receiver;
- Airborne's shielded UAV;
- ARVA transmitter.

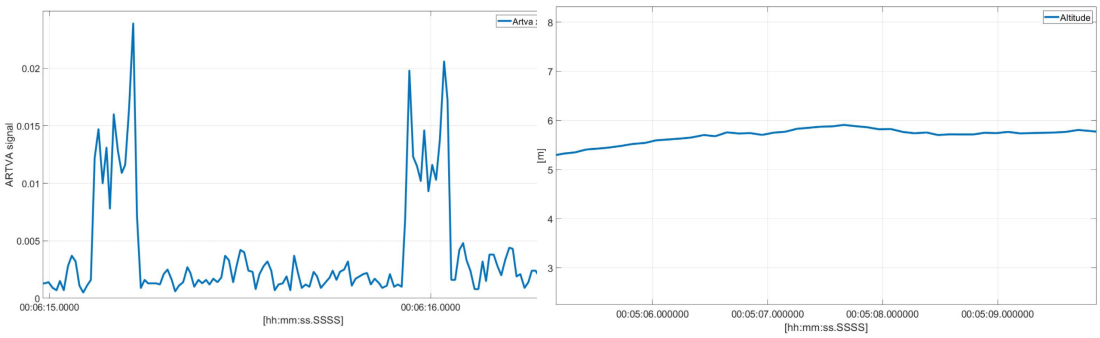
The ARVA data used are the three-power related signals coming from the payload integration port of the UAV. Since the readings of the antenna are correlated to the position and orientation of the UAV with respect to the transmitter, the UAV telemetry has been synchronized with the ARVA data using a ground-truth signal coming from the RC. In particular, data concerning the position and orientation of the UAV are used during the training phase. The signals coming from the three-axis of the ARVA receiver $x, y, z \in \mathbb{R}^3$ are used as input in the network. The altitude of the UAV $h \in \mathbb{R}$ is used as a fourth input obtaining the following input vector:

$$U = \begin{bmatrix} x \\ y \\ z \\ h \end{bmatrix} \mathbb{R}^4 \tag{4.3}$$

The network outputs the x-distance and y-distance with respect to the UAV local reference frame.

$$Y = \begin{bmatrix} x_{bf} \\ y_{bf} \end{bmatrix} \in \mathbb{R}^2 \tag{4.4}$$

The dataset for training and testing has been collected by performing some well-known search patterns described in section 2.1.1 (Greek, triangular, sawtooth grid) with different yaw angles. The relative altitude has been kept fixed during the search pattern, as in a real SAR operation. Additionally, some data have been generated performing free flights. The flights have been taken on two different days for a total of almost 40 minutes of flight time. The dataset has been augmented to avoid overfitting. The augmentation has been performed by shifting the time series using time-domain methods as in [49].



(A) (B)

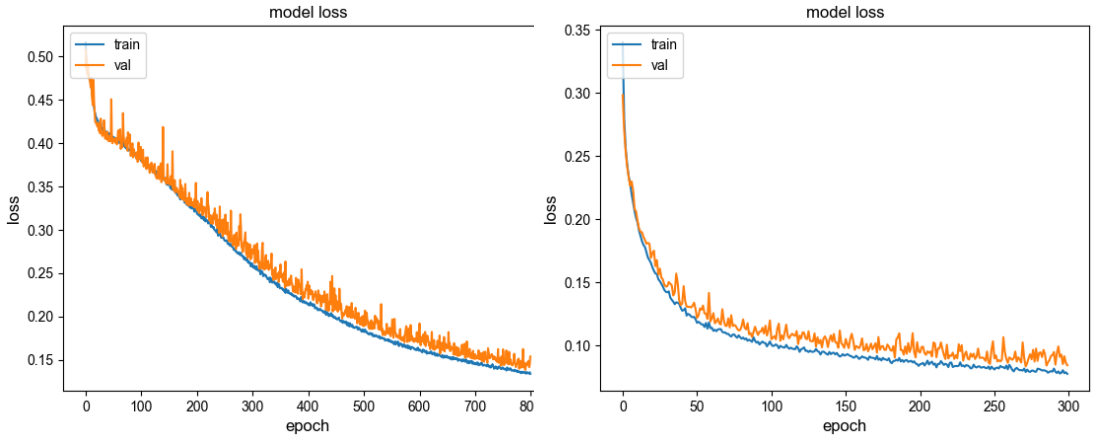
FIGURE 4.10: *Examples of input signals of the RCNN. In* (A) *The readings of the X-axis of the antenna when there is a beacon present. The presence of the two pulses is a clear sign of the presence of the transmitting beacon. In* (B) *the altitude of the quadrotor during the testing operation.*

The inputs are time series of 200 samples, this leads to an ARVA signal of 2 seconds. This means that the network has to acquire the inputs for 2s before giving

the output. This acquisition time is necessary because the transmitting beacon emits pulses of 70ms every 1000ms with a variance of 300ms. As described in section 4.2.2, a bigger acquisition time leads to better identification but it will slow down the search speed. Here we selected a time window to acquire at least one pulse in the worst case and two pulses in the best case.
The data acquired during the flight are signals with different sampling times. In order to use the data coming from the drone telemetry and correlate them with the ARVA signals, there is the need to obtain a common timestamp. Consequently, the data with a sample time $t_c \leq 100Hz$ have been interpolated using splines to be consistent with the ARVA data. Finally, the outputs of the network x_{bf}, y_{bf} have been normalized. Since the *Home* position of the quadrotor is coincident with the position of the transmitter, through the telemetry data is possible to check the real distance from transmitter and receiver and express in body frame using the attitude data. Therefore, the datasets have been labelled and divided into training (75%), validation (20%) and test (5%) sets.
We tried different networks thanks to the support given by the architecture of [20]. In the end, as already pointed out, we decided to use a Multi-Output Multivariate Regressive CNN because of the good performance registered. The network is composed of the first layer of 80 fully connected nodes and two convolution layers. The fourth input h is used in the first convolution layer. The idea is to have the fully connected layer handle just the ARVA data. The successive convolution layers extract the information from the ARVA signals and the altitude signal. We selected *Adam* as an optimizer and used to following loss function:

$$Loss_{RMSE} = \sqrt{1/N \sum^{N} \left(Y_{true} - Y_{pred}\right)^2} \tag{4.5}$$

where, $Y_{true} \in \mathbb{R}^2$ is the vector of the measured outputs, $Y_{pred} \in \mathbb{R}^2$ is the vector of the predicted outputs. N is the total number of samples of the time series.
In this work, given the regressive nature of the outputs, we intend the accuracy of our network as the cross-correlation between the predicted outputs and the measured outputs. Using a batch size $b_s = 16$, the predicted signal is 91.4% similar to the real signal. The loss function during training and validation can be seen in Fig. 4.11. Since the datasets have been created in two different experiments, transfer learning has been used to improve the results of the first dataset. In fig. 4.12 the performances during the testing procedure are shown. The signals represent the distances between the ARVA receiver and the ARVA transmitter, normalized.



(A) (B)

FIGURE 4.11: *Loss graphs of the training and testing procedures. In* (A) *the loss during the training and validation with the first dataset. In* (B) *the Loss during the training and validation with the second dataset and with transfer learning.*

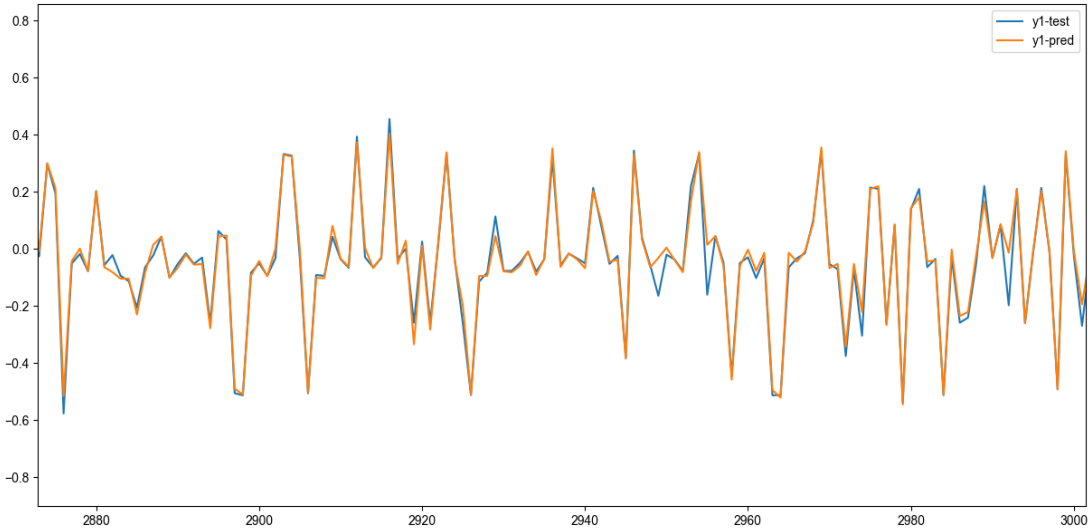


FIGURE 4.12: *Predicted distances versus measured distances during the testing procedure. The results are normalized between 0 and 30m.*

The results are encouraging in the development of an online algorithm. The saturation of the ARVA signal when the antenna is in the proximity of the transmitter can cause some errors when under 5m from the transmitter, as can be seen from fig. 4.12. This can be solved, and the performances can be further improved, by extending the dataset. We aim to fully develop the algorithm in a real-time scenario and extend it by adding multi ARVA recognition since is the typical case in real SAR operations. The repository with the dataset can be found in [52]. Is worth stressing the fact that, the implementation of a network to handle the ARVA data has been made possible by the usage of the Airborne shielded UAV and its integration with the 3D ARVA payload.

5 Navigation and Path Planning Strategies

The three major components in automated vehicles movement are guidance, navigation, and control. One fundamental part that falls under the guidance, and is intrinsically connected to navigation, is path and trajectory planning. Independently on the application, the goal of a path planner is to generate a suitable trajectory for the automated vehicle. In particular, in this work, we refer to UAVs. The generation of such a trajectory is a challenging task because it has to take into account knowledge of the state of the UAV and the surrounding environment, the dynamic differential constraints of the UAV, the presence of external disturbances, and in the case of SAR operation the state of the payload. Additionally, in SAR there is the need to run it in real-time using the onboard computation. The literature is full of strategies that allow finding global and optimized trajectory that satisfies different kinds of constraints [40] [25][7][38] but might be unsuitable for a SAR application where local modification of the trajectory might be occurring. In this work, the contribution of path and trajectory planning is the development of a method suitable to SAR applications that guarantee kinodynamic feasibility for a UAV, obstacle avoidance and replanning. Besides, we present a waypoints generation method applied in search operations with no pre-knowledge of the map.

5.1 Waypoint Generator

Automatic flight patterns are an essential feature in SAR operation. In fact, when carrying a payload such as ARTVA or RECCO, or infrared cameras, an automatic flight pattern can help to reduce the task assigned to the pilot. Consequently, the pilot can put more attention in monitoring the state of the payload, searching for the detection of a victim. Besides, they provide repeatable and more reliable search operations, avoid the risk of human error during difficult flights when there are harsh meteorological conditions and long-distance patrolling. In Airborne, the automatic flight pattern developed takes into account the followings:

- Integration with the payloads search strategies;
- Absence of a ground station and other devices than the RC;
- Quick deployment of the algorithm in unknown maps;

(A)

(B)

FIGURE 5.1: *An example of application of the automatic search pattern performed during a flight test by CAI. In* (A) *the top view of the entire path, where can be seed the execution of 3 different greek patterns. In* (B) *the side view of the entire path, where can be seen the terrain-following feature of the algorithm that keeps the altitude from the terrain constant, in order to better use the payloads.*

- Possibility to change in real-time the path to better patrol the search area.

Following the search methods presented in section 2.1 we developed a waypoints generator that can reproduce the greek pattern, the triangular pattern, and the saw-tooth pattern. The pilot can choose the grid type and can start the execution of the path directly from the RC. Initially, the UAV hovers on a certain position that, through an RC command, is acquired by the algorithm. Consequently, the UAV has to be moved by the pilot to the second point of the grid that, through the same RC command, is acquired by the algorithm. The line delimited by the two points is the starting line of the grid and is used to define the heading of the pattern. The successive points are entirely calculated by the algorithm based on pre-determined parameters like the line-to-line distance and the altitude of the grid. Based on the length of the first track the pilot can tune the extension of the grid to better fit the research area. Additionally, the heading of the pattern can be changed in real-time using the Yaw axis of the RC. This allows a further adaptation to irregular search areas like the post avalanche scenarios. During the whole operation, the pilot has to manoeuvre the drone only on the first two points of the grid, the rest is performed

autonomously. In this way, the pilot can focus on reading the data coming from the payload.
Let's define $\gamma_1, \theta_1 \in \mathbb{R}$ as the two coordinates of the first waypoint Wp_1, and $\gamma_2, \theta_2 \in \mathbb{R}$ as the coordinates of the second waypoint Wp_2. Then, we can calculate the azimuth angle $A \in \mathbb{R}$ of the line that pass between Wp_1 and Wp_2.

$$\alpha = \arccos[\cos\gamma_2 \cos\gamma_1 + \sin\gamma_2 \sin\gamma_1 \cos\theta_2 - \theta_1]$$

$$\bar{A} = \arcsin \frac{\sin\gamma_2 \sin\theta_2 - \theta_1}{\sin\alpha} \tag{5.1}$$

where $A = \bar{A} \pm \pi/2$.

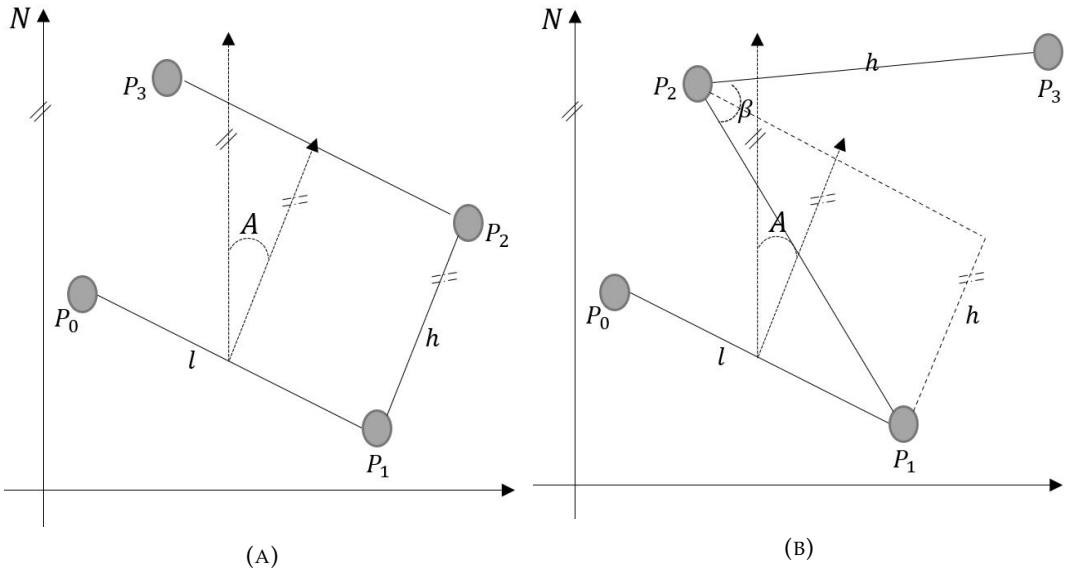


FIGURE 5.2: *Examples of the creation of two different types of grid patterns. In* (A) *the creation of the greek pattern. In* (B) *the creation of a triangular pattern. Is worth noticing that based on the value of β we can obtain the sawtooth pattern.*

Defining $l \in \mathbb{R}$ as the distance between the first two waypoints (i.e. the width of the grid) and $h \in \mathbb{R}$ as the line-to-line distance, we can characterize completely the grid and compute all the waypoints autonomously by calculating the azimuth angle of the initial segment and, consequently, the heading of the grid. During the execution of the grid, the action of the Yaw axis of the RC modifies azimuth A, leading to a change in the grid heading.
The grid generation algorithm is implemented as an additional module into the PX4 flight controller's firmware. Consequently, it is possible to easily send the coordinates of the waypoints to the autopilot in order to generate a trajectory that passes through the waypoints and has some constraints respected. The flight controller generates symmetric smooth S curves with bounds on jerk and acceleration. It is possible to bound also the max translation speed based on the payload acquisition

time. The waypoints are considered reached when the following condition is fulfilled:

$$||p_{UAV}(t) - Wp_j||_2 \leq \delta \tag{5.2}$$

where, $p_{UAV} \in \mathbb{R}^3$ is the position of the UAV at a certain time t, $Wp_j \in \mathbb{R}^3$ is the position of the j-th waypoint, and δ is the acceptance radius.

Algorithm 2 Grid generation algorithm

Require: $\gamma_0, \gamma_1, \theta_0, \theta_1, N, h$
Ensure: $i = 1$
Ensure: $A = \bar{A} \pm \pi/2$
Ensure: $type = k \in [0, 1, 2]$
  while $i \leq N$ **do**
    if $type = 0$ **then**
      $\gamma_{i+1} = \gamma_i + h \cos A$
      $\theta_{i+1} = \theta_i + h \sin A$
      $\gamma_{i+2} = \gamma_{i-1} + h \cos A$
      $\theta_{i+2} = \theta_{i-1} + h \sin A$
    end if
    if $type = 1$ **then**
      if $i == 1$ **then**
        $\gamma_{i+1} = \gamma_{i-1} + h \cos A$
        $\theta_{i+1} = \theta_{i-1} + h \sin A$
        $\gamma_{i+2} = \gamma_i + 2h \cos A$
        $\theta_{i+2} = \theta_i + 2h \sin A$
      else
        $\gamma_{i+1} = \gamma_{i-1} + 2h \cos A$
        $\theta_{i+1} = \theta_{i-1} + 2h \sin A$
        $\gamma_{i+2} = \gamma_i + 2h \cos A$
        $\theta_{i+2} = \theta_i + 2h \sin A$
      end if
    end if
    if $type = 2$ **then**
      $\gamma_{i+1} = \gamma_{i-1} + h \cos A$
      $\theta_{i+1} = \theta_{i-1} + h \sin A$
      $\gamma_{i+2} = \gamma_i + h \cos A$
      $\theta_{i+2} = \theta_i + h \sin A$
    end if
    i = i+2
  end while

5.2 Terrain Following

In Airborne, the SAR operations, use the payloads to perform a patrolling of a certain area, unknown a priori. In order to maintain a certain consistency in the usage of the payloads, the same altitude from the ground must be kept. This becomes a challenging task when the UAV operates in environments (i.e. post avalanche scenarios) with sudden terrain altitude changes, with the presence of big objects on the

terrain, or with harsh and irregular slopes. Additionally, if the UAV is teleoperated over long distances, it becomes really hard to distinguish by eyes if the drone is losing or gaining altitude from the terrain. The implementation of a terrain following method becomes of fundamental importance in guaranteeing a certain level of consistency during the SAR operation and preserving the safeness of the machine over long-distance flights.
The autopilot used in the AlCan02 presents a terrain following feature that uses the distance extracted from a lidar sensor. Even if this could be useful in flights with almost flat terrain, or low speed operation, is not suitable to be used in a typical alpine SAR operation. In order to extend the functionality of the terrain following feature of the autopilot in the application case in Airborne, we implemented a 3 lidar system within the AlCan02. The first lidar points downward and is placed under the Autopilot module. The others are placed at a distance d_l along with the autopilot module, tilted by $\alpha = 60°$. The data coming from the three lidars are fused

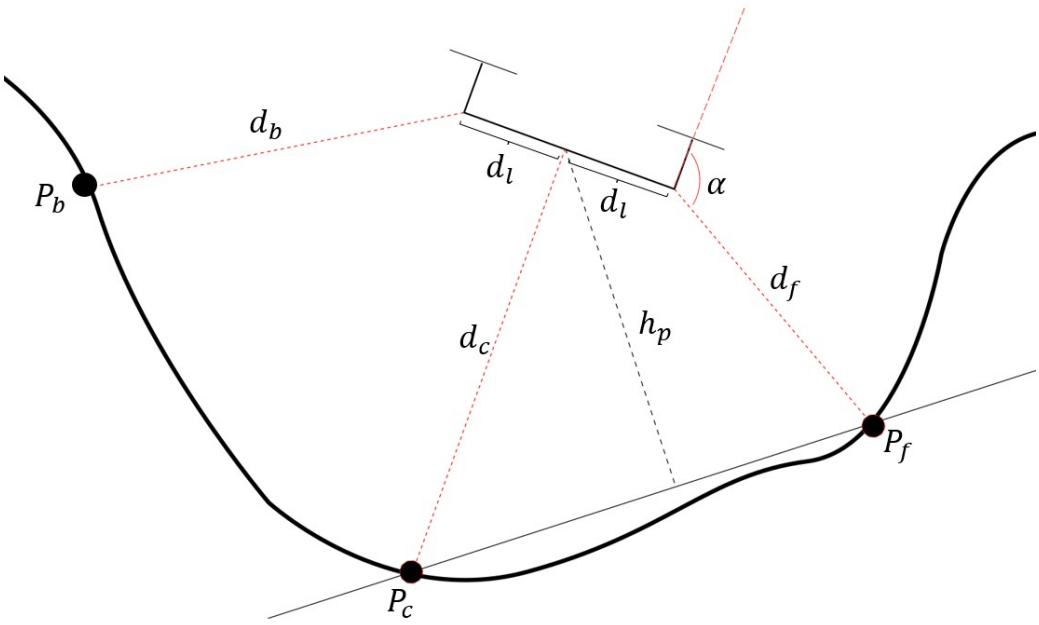


FIGURE 5.3: *The representation of the three lidars system implemented to enhance the terrain following features of the autopilot.*

to obtain an estimate of the distance from the terrain, even if there is a huge slope. Consequently, is possible to maintain a certain distance from the terrain even at a high speed and even if the terrain is irregular. The estimated distance h_p is sent to the autopilot to be used as ground distance data in the terrain following algorithm. let's define $V \subset \mathbb{R}$ as the set of valid values of the lidars, θ as the pitch angle of the quadrotor, $v_x, v_y \in \mathbb{R}$ as the horizontal translation velocities of the quadrotor, α as the angle between the body reference frame's vertical axis, and the tilted lidars' directions, d_f, d_c, d_b as the readings of the lidars. Then, we can express the lidar fusion algorithm as indicated in algorithm 3. The data of the lidars are first to verify the integrity of the reading and if is a valid value belonging to$V \subset \mathbb{R}$. Consequently, the

Algorithm 3 Three lidar integration algorithm

Require: $d_f, d_c, d_b, d_l, \alpha, \theta, v_x, v_y$
Ensure: $d_f, d_c, d_b \in V$
  for each $d_f, d_c, d_b \in V$ **do**
    if $d_f < d_b$ **then**
      $P_f[1] = d_l + d_f \sin \alpha$
      $P_f[2] = -d_f \cos \alpha$
      $P_c[1] = 0$
      $P_c[2] = -d_c$
      $m = \frac{P_f[2]-P_c[2]}{P_f[1]-P_c[1]}$
      $b = P_f[2] - m * P_c[1]$
    else
      $P_b[1] = d_l + d_b \sin \alpha$
      $P_b[2] = -d_b \cos \alpha$
      $P_c[1] = 0$
      $P_c[2] = -d_c$
      $m = \frac{P_b[2]-P_c[2]}{P_b[1]-P_c[1]}$
      $b = P_b[2] - m * P_c[1]$
    end if
    $P_m[1] = -\frac{bm}{m^2+1}$
    $P_m[2] = \frac{b}{m^2+1}$
    if Velocty Correction **then**
      $P_m[1] = P_m[1] + Kv_x \cos \arctan m$
      $P_m[2] = P_m[2] + Kv_x \sin \arctan m$
    end if
    $h_p = \frac{|\sin \theta P_m[1] + \cos \theta P_m[2]|}{\sqrt{\sin \theta^2 + \cos \theta^2}}$
  end for

line between the closest two points determined by the lidars is calculated. We can think of this line as the line that follows the slope of the terrain. In the case of more pronounced terrains, the inclination of the line is more pronounced. The position of the quadrotor is projected on these lines and moved based on a velocity correction. This helps to compensate for the tilting of the quadrotor in the readings of the lidars. Finally, the distance between the UAV and this point is calculated. Once established a certain altitude to be kept during the search manoeuvres (i.e. during the execution of the automatic flight patterns in section 5.1) it can be kept constant by acting on h_p. It is worth stressing the fact that, by the usage of the three lidars data, it is possible to keep the same altitude from the ground even in the case of steep hills or soils with overhangs.
The lidar fusion algorithm is implemented inside an Arduino Pro mini within the autopilot module. It communicates the data h_p with the reading of the lidars to the autopilot, through a serial link, to perform the terrain following feature.

FIGURE 5.4: *A post avalanche scenario where the AlCan02 was tested. The irregular terrain makes it difficult and, if too close to the ground, even dangerous to perform terrain following with a single lidar.*

5.3 Trajectory Generator

Path planning is becoming more and more important as robot usage is spreading. Motion planning in a complex environment plays a key role in accomplishing tasks assigned safely and reliably. Over the past decade, the request for autonomous navigating UAVs is grown in many different applications like SAR, patrolling, and industrial applications. A guidance algorithm that allows autonomous path generation for the multicopter can alleviate the workload of the operator who can focus the

attention on the mission rather than piloting the UAV. However, the generation of trajectory is a challenging issue, since it needs to take into account the dynamic constraints of the aircraft and the obstacles. Among all the trajectories that satisfy the constraints, we might want to find an optimal one. This arises an optimization problem that leads to heavy computational load tasks. Moreover, we want to allow an online computation of the trajectory, also in case of environmental changes, which requires a quick computation and a fast replanning algorithm. Basically, trajectory planning consists in finding a relationship between time and space through a parametric function that correlates the position at each time instant. There is a huge set of categories that can define the particular trajectory used. One of the first main distinctions is to classify the trajectories as one- or multi-dimensional. It naturally follows that the one-dimensional trajectories are designed for one degree-of-freedom system, while the multi-dimensional ones are designed for multidimensional systems. The categorization can be deeper, as an example, we might be interested in finding a single-dimensional trajectory for a 3D system that is constrained on a single axis. Additionally, trajectories can be further classified depending on the fact that the motion is defined by taking into consideration a set of intermediate waypoints to be interpolated or just by taking some initial and final points. Is worth noticing that,

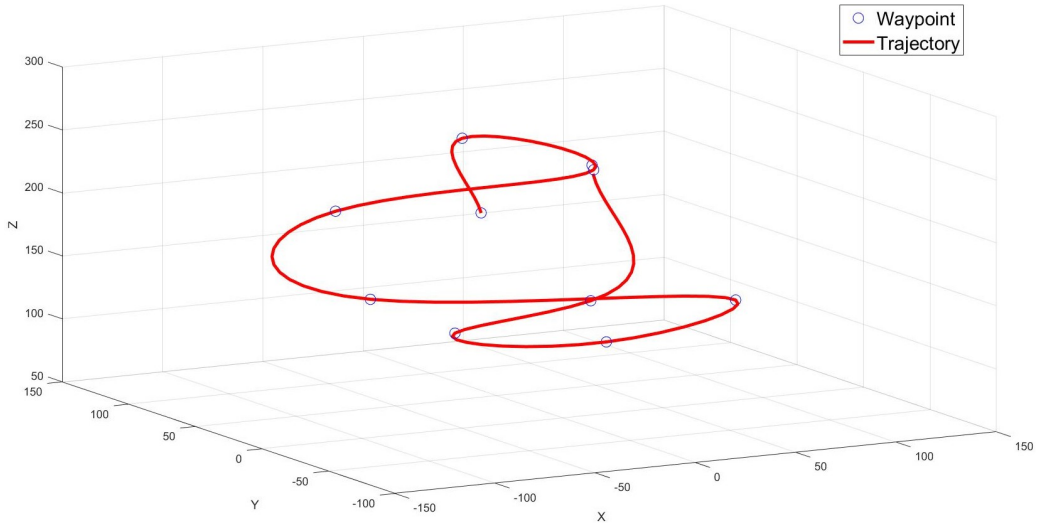


FIGURE 5.5: *An example of a multipoint trajectory obtained using splines and interpolation of the waypoints.*

the multipoint trajectories can just approximate the waypoints, meaning that the trajectory doesn't have to exactly interpolate them but can pass in a neighbourhood of the waypoints. An example of such a multipoint approximated trajectory is the grid generation algorithm in section 5.1, where the trajectory generated by the autopilot doesn't have to interpolate exactly the waypoints coming from the algorithm 2but is considered found when 5.2 is fulfilled. Since in this work we intend to generate trajectories for UAVs that move in an unknown map with possible obstacles, we use multi-dimensional multipoint trajectories. Additionally, given the application

to SAR operation and the possible presence of obstacles, we want an exact interpolation of the waypoints.

Definition 5.3.1 (Configuration Space)**.** A set of variables X that completely describes the state of a UAV is called configuration. The set of all possible configurations is called configuration space and denoted $\mathbb{S}$. A particular configuration X_{tot} in which the UAV is colliding with the environment or with itself is called colliding configuration. The subset $\mathbb{S}_{free} \subseteq \mathbb{S}$ of all non-colliding configurations is called free space.

Usually, the goal in defining a trajectory is to bring the starting configuration X_{start} to a final configuration X_{end} staying in $\mathbb{S}_{free}$, and respecting some constraints. Both initial and final configurations are usually known.

Definition 5.3.2. Given $u \in [0,1]$, a path Γ_p is a continuous function $\Gamma_p : [0,1] \rightarrow \mathbb{S}_{free}$ that connects X_{start} and X_{end} whose generic element is indicated as $X(u)$ and such that $X(0) = X_{start}$, $X(1) = X_{end}$.

Definition 5.3.3. Given a path Γ_p, and a continuous strictly increasing function $f : [0.T] \rightarrow [0,1]$ with $f(0) = 0$, and $f(T) = 1$, a trajectory is defined as $\Gamma : [0,N] \rightarrow \mathbb{S}$ that connects X_{start} and X_{end} whose generic element is indicated as $X(f(t))$ with $t \in [0,T]$ and such that $X(0) = X_{start}$, $X(T) = X_{end}$.

Generally speaking, it is worth noticing that, we could design $\Gamma : [0,N] \rightarrow \mathbb{S}$ in applications where collisions are tolerated, or needed. In our setup, we always refer to the safe space $\mathbb{S}$ in order to guarantee no collision with the environment.
From definition 5.3.3 is clear that we can think of a trajectory as a geometric path with a motion law $u = f(t)$ assigned. The determination of the constraints imposed by a certain task determines, separately, the selection of the motion law and the geometric path. In fact, the motion law could be determined by the imposition of conditions on the maximum velocities, derivatives, jerk and snap, or torques that the actuation system can provide. Conversely, the geometric path could descend from the safeness of the path or from some optimal logic (i.e. the maximum distance from obstacles, or the minimum distance between waypoints). Attention must be drawn to the fact that, based on the particular algorithm developed, we might want to impose constraints that influence both geometrical path and time law.

5.3.1 Trajectory Planning Problem

As already pointed out, the trajectory planning problem consists in determining a succession of configurations that connects an initial set of states X_{start} to a goal configuration X_{end} respecting some constraints. The constraints might include conditions on the maximum velocity allowed, restriction of the patch in $\mathbb{S}_{free}$ and so on. The challenges in building a good planner are many. In this work, we consider some of the major ones in the following.

Safeness

In scenarios in which no collisions are allowed, safeness refers to the ability of the algorithm to generate a trajectory in no collisions areas, in other words, that guarantees $\Gamma : [0, N] \rightarrow \mathbb{S}_{free}$. Despite there exist methods that act on the dynamics of the vehicle to guarantee no collisions, for example by stopping the motion of reducing the speed, in planning the trajectory safeness is a low-level mandatory requirement that has to be fulfilled independently of the high-level safe procedures that might be implemented. The respect of the safeness of the trajectory does not depend only on determining a path that physically avoids obstacles, but depend also on the feasibility constraints.

Feasibility

Feasibility refers to the ability of the planner to generate trajectories that are feasible to be executed by a particular vehicle selected. Consequently, they have to be feasible for the dynamic model of the chosen system. Let's define $x_r(t) \in \mathbb{R}^n$ as the reference state trajectory, and $u_r(t) \in \mathbb{R}^m$ as the input reference signal of a system. Define a general non linear model of the dynamic of the chosen vehicle as:

$$\dot{x} = f(x(t), u(t)) \tag{5.3}$$

where, $f : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^n$ a locally Lipschitz function, and $x \in \mathbb{R}^n, u \in \mathbb{R}^m$. Then, we can introduce the following.

Definition 5.3.4. Given $x_r(t) \in \mathbb{R}^n$ and $u_r(t) \in \mathbb{R}^m$ the reference trajectory $x_r(t), u_r(t)$ is dynamically feasible for a system like 5.3 if

$$\dot{x}_R(t) = f(x_r(t), u_r(t)) \tag{5.4}$$

Where f is the same function characterizing 5.3.

The definition 5.3.4 means that by selecting a reference input $u_r(t)$, the state of the system evolves as a possible trajectory of the system 5.3 and so it respects all the dynamics constraints of the particular system selected. Dynamic constraints are one of the major factors that increase the computational complexity due to differential, complex, and non linear equations. For this reason, for the sake of simplicity of the implementation, they are often simplified. Besides, respecting these types of constraints is of paramount importance in trajectory planning. In fact, if not satisfied, is not possible to guarantee an exact tracking of the generated trajectory. It follows that even if the trajectory lives in $\mathbb{S}_{free}$, the vehicle could assume a set of configurations, in $\mathbb{S}$, that leads to a possible collision with obstacles.

Optimality

In determining a trajectory we might want to optimize some objective function. The optimization is usually given in terms of maximization or minimization of the objective function. Based on different applications we might be interested in obtaining different trajectories that are optimal in different senses. As an example, in a long-distance SAR, we might want to obtain a trajectory that minimized energy consumption. Conversely in a patrolling operation with the camera as a payload, we might want a smooth trajectory that minimizes the snap.
In determining a procedure to optimize the trajectory generated by the planner we also add constrains to be respected that guarantee the safeness 5.3.1 and dynamic feasibility 5.3.1. The trajectory planning problem becomes a constrained optimization problem that can be expressed as follows:

$$\begin{aligned} \min_{\chi} \quad & J(\chi) \\ \text{s. t.} \quad & A_{eq}\chi = b_{eq} \\ & A_{ineq}\chi \leq b_{ineq} \\ & \chi \in \mathbb{S}_{free} \end{aligned} \tag{5.5}$$

where, χ is a vector of variables representing our degrees of freedoms in the optimization procedure, $J(\chi)$ is the objective function to be minimized with respect to χ. The constraints are represented as inequality constraints and equality constraints. The equality constraints usually are used to fix some relationship between the optimization variables, besides they could be used to fix some of those variables. As an example, we might want to impose the initial and final values of χ. The inequality constraints restrict the solution space to a subset, based on the particular relation implemented. We could use inequality constraints to impose bound on the values of the derivatives in order to obtain a trajectory that is feasible for a certain vehicle. It is worth noticing that, the optimization problem could also be expressed in terms of maximization of a certain objective function.
Based on the particular objective function selected and on the number and types of constraints used, optimal trajectory planning can be a very challenging task and computationally expansive [13][28].

Additional Proprieties

The choice of the planner is not only determined by optimality, safeness, and feasibility of the trajectory. We might want some additional proprieties based on the particular application in which we want to apply the planner. In a real-time application, like the Airborne's one, we are also interested in keeping as low as possible the run-time of the planner. In fact, since it is strictly related to how fast is possible to act on a change in the environment, having a low computational time allows one to use it in a real-time application where an obstacle can pop up, or the map is modified

by some external factor. The run-time is given by the particular algorithm used and the constraints considered. As an example, for discretized and graph-search algorithms the fine discretization might lead to high dimension graphs and high computational complexity, while in a randomized algorithm the complexity can come from the nodes' connections based on kinodynamic constraints. Generally speaking, for optimal algorithms, the complexity is given by the choice of the objective function and the numbers and types of constraints.
Strictly linked to the run-time there is the replanning feature. In this work, we define replanning as the ability of a trajectory planner to modify a part or totally, the trajectory after the first computation. This ability is strictly related to the particular planner used. In fact, some planners might be unable to compute the new trajectory at any time (i.e. grid-based planners). We want to stress the fact that, with good replanning capabilities, it is possible to modify a part of the trajectory without the need to recompute the whole procedure. This comes in handy when we detect a new obstacle and we just want to modify a part of the trajectory in order to avoid it. It naturally follows that an algorithm with an incredibly low run-time could recompute the whole trajectory even if it needs small changes. The point in having replanning is that the amount of calculation to modify a certain part of the trajectory is lower with respect to recomputing the whole process. This is true especially for applications, like patrolling, in which we have a big precomputed trajectory that we want occasionally to modify to avoid unpredicted obstacles. This is related to the concept of globality of the planner. The globality features refer to the set of configurations in which the planning algorithm is computed and valid. Given a set of configurations $X_{tot} \subseteq \mathbb{S}_{free}$ the planner is said to be global if it runs on X_{tot}, otherwise is said to be local if it runs on $X_{loc} \subset X_{tot}$. It follows that the best algorithm should be global. In fact, they provide a trajectory with global proprieties in a known map, but they are not suitable to be applied in every situation. Besides, the increased state-space leads to an increased computational time. Furthermore, in the case of application in unknown environments the globality is not useful because the motion planner will evaluate only local information, making it impossible to plan a trajectory outside the sensors range. On the other hand, the local approach is not able to guarantee the performance of global methods but it has reduced complexity and can be used even if the map is not fully known. In fact, local algorithms are usually implemented on a subset of configurations given by the exteroceptive sensors of the vehicle. In order to reach a trade-off between local and global performance, many approaches use a global planner associated with a local planner that processes the trajectories in the proximity of the vehicle.

5.3.2 Kinodynamic Optimal Trajectories Generation

We developed a strategy for trajectory planning for quadcopters using Bernstein polynomials. The focus is on generating trajectories that are functionally controllable for this class of under-actuated vehicles and that guarantee a safe path to

avoid obstacles. The algorithm is suitable for local replanning and onboard quick re-computation of the trajectory when unplanned obstacles pop up. Bernstein polynomials are particularly suited to integrate all the constraints into a quadratic programming problem that can be solved in polynomial time. We consider the motion planning module split between a safe waypoint generator and a trajectory through-waypoint generator. In this work, we focus on the latter. The waypoints can be generated by different methods, some suppose a map to be known and outputs the positions of waypoints, like RRT developed by LaValle [32] or RRT* PRM* and RRG proposed by Karaman and Frazzoli [29]. Other methods allow finding directly a curve connecting two points in a complex map. An interesting approach of his kind is the Fast Marching method [45], a numerical method used for simulating a propagation of a wavefront. Besides, this method has been used to help to find safe areas in which build the trajectory using Bezier curves [25]. Over the past decades, many approaches to generate curves through waypoints have been proposed likes polynomial methods [38] [7]. Similarly, Bernstein polynomials can be used to address this problem. The work of Constanzi and Fanelli [18] uses a Bezier curve to address an optimization problem that, solved with a direct search method, outputs the control point of the Bezier curve. The Bernstein polynomials are also used to perform collision avoidance [37] using a time parametrization and calculating a detour curve to be added to the original curve to perform the avoidance manoeuvre. Recent work [16] shows how is possible to perform optimal motion planning using Bernstein's approximants in real-time that guarantees safety in a complex environment for a multiple vehicle mission.
In this work, we calculate a trajectory composed of optimal Bézier curves and we guarantee the 4-times differentiability of the curve, replanning feature, and the respect of the derivative bounds imposed by the dynamics of the quadrotor.

Problem Formulation

The trajectory generated must take into account the quadrotor's dynamics, in order to be tracked by the quadrotor. In doing so, we suppose to know the model of the quadrotor and so, his dynamics. A mathematical model for this kind of vehicle can be derived using the Newton-Euler equations of motion of a rigid body in space configuration $\mathbb{R}^3$. By assuming that the body frame has its axis aligned with the principal axis of inertia of the rigid body and considering $F_c = \{O_c, i_c, j_c, k_c\}$ as the inertial coordinate frame, we can express the dynamical model of the quadrotor, with respect to the inertial frame, as:

$$\begin{aligned} M_m \ddot{p} &= R_\psi f_c + M_m g e_3 + f_d \\ J_m \dot{\omega} &= -\omega \times J_m \omega + \tau_c + \tau_d \end{aligned} \tag{5.6}$$

where M_m is the total mass of the quadrotor, J_m is his inertia, $p = [x, y, z]^T$ is the position of the center of mass, ω the angular velocity in body frame, and finally e_3

is the unit vector $e_3 = [0, 0, 1]^T$. The resulting force and torque generated by disturbances are represented, respectively, by f_d and τ_d. Let's indicate $u := [u_1, u_2, u_3, u_4]^T$ as the vector of all the force-torque components produced by the 4 actuator modules of the quadrotor. We indicate the control force and control torque vector as $f_c \in \mathbb{R}^3$ and $\tau_c \in \mathbb{R}^3$ applied to the centre of mass. By defining ψ_i as the angle by which the reference frame should be rotated around the z-axis to align the x-axis with the one of the body frame, we can indicate R_ψ as the rotation matrix relating reference frame and body frame, as follows:

$$R_\psi := \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since the quadrotor dynamics is deferentially flat [48], the state and the inputs can be written as an algebraic function of flat outputs and their derivatives. The flat outputs we chose are given by

$$\Gamma = \begin{bmatrix} x_m, y_m, z_m, \psi \end{bmatrix}^T,$$

where $[x_m, y_m, z_m]$ the coordinates of the center of mass in $F_c = \{O_c, i_c, j_c, k_c\}$ and ψ the yaw angle. Is then possible to define the trajectory as a curve in space of flat outputs:

$$\Gamma(t) : [T_0, T_f] \rightarrow \mathbb{S}_{free} \subset \mathbb{R}^3$$

Consequently, we can relate bounds on the derivatives of the flat outputs to the inputs. This allows us to bound the derivatives of the curve, based on the bounds in the quadcopter dynamics. In our work we leave the Ψ as an additional degree of freedom of the quadcopter, that can eventually be imposed by other algorithms, i.e. ROI (region of interest) tracking with a camera within gimbals.
Given $\gamma^{(i)}$ as a bound value of the i^{th} derivatives of the quadrotor model, we can write:

$$-\gamma^{(i)} \leq \Gamma(t)^{(i)} \leq \gamma^{(i)}$$

Therefore, if the following relation if fulfilled for all desired grade i and for all $t \in [T_0, T_f]$ the generated trajectory can be followed by the selected quadrotor i.g. is dynamically feasible. It is worth noticing that, the bounds we are considering are not the real bounds on the actuators. In fact, they don't take into account the saturation of the single actuators, nor the discontinuity at the input, Nonetheless, we are considering a good approximation that let us obtain constraints that can be easily implemented in the optimization problem.

Given a map of the environment, the vehicle has to pass through m waypoints

$$Wp_j \in \mathbb{R}^3 = \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} \quad j \in [0, m-1]$$

where Wp_0 and Wp_{m-1} are the start and endpoint respectively, and where Wp_j with $j \in [1, m-2]$ are intermediate waypoints. The waypoints are generated from a high-level algorithm as [32] or [30], [6]. We also consider that the waypoints are not fixed, but they may change over time and their number might vary because the map is considered dynamic, i.e. new obstacles are detected or can move. For this scenario, we need the trajectory to be replanned quickly in case of a change of obstacles or waypoints during the execution of the path. Summarizing, we are interested in finding a trajectory $\Gamma(t) \in \mathbb{R}^3$, which is a position trajectory for the quadrotor, with the following proprieties:

- The trajectory $\Gamma(t)$ passes through all the Wp_j;
- The trajectory $\Gamma(t)$ should allow re-planning;
- The trajectory $\Gamma(t)$ is dynamically feasible for a quadrotor;
- The trajectory $\Gamma(t)$ is optimal w.r.t. a cost function;
- The trajectory $\Gamma(t)$ avoids the obstacles in the map.

The trajectory is considered as a piecewise trajectory composed by $m-1$ segments. Each segment is a Bézier curve interpolating a pair of consecutive waypoints. Considering a parameterization function $u_j = f_j(t_j)$ with $t_j \in [T_j, T_{j+1}]$ for $j \in [0, m-2]$. The trajectory can be written as follows:

$$\Gamma(t) = \begin{cases} \sum_{i=0}^{n} b_i^n \left(f_0(t_0)\right) \cdot P_0^i & t_0 \in [T_0, T_1] \\ \sum_{i=0}^{n} b_i^n (f_1(t_1)) \cdot P_1^i & t_i \in [T_1, T_2] \\ \quad \vdots & \quad \vdots \\ \sum_{i=0}^{n} b_i^n (f_{m-2}(t_{m\text{-}2})) \cdot P_{m-2}^i & t_{m-2} \in [T_{m\text{-}2}, T_{m\text{-}1}] \end{cases} \tag{5.7}$$

where P_j^i are the control point of a Bézier curve of order n with $i \in [0, n]$. In order to build the trajectory, we are interested in finding P_j^i and a time parameterization $f_j(t_j)$. The complete trajectory must satisfy the constraints discussed previously.

For the choice of $f_j(t_j)$ we make the following assumptions: The function is of class C^1 and the parametric function is monotonically increasing and $f_j(t_j) \in$

$[0,1]$.We use the following parametric function for the generic j^{th} tract :

$$f_j(t) = \frac{t_j - T_j}{s_j} \tag{5.8}$$

Where s_j is the time span associated to the j^{th} tract, $T_j = T_{j-1} + s_{j-1}$ and $f_j(t_j) \in [0,1]$ *with* $t_j \in [T_j, T_j + s_j]$.
The singles Bézier curves are composed using Bernstein polynomial basis defined as:

$$b_i^n(u) = \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \tag{5.9}$$

Where n is the degree of the basis, $i \in [0,n]$, $\binom{n}{i}$ is the binomial coefficient, and $u \in [0,1]$.
It is now possible to define a Bézier curve as following:

$$B_j(u) = \sum_{i=0}^{n} b_i^n(u) \cdot P_j^i \tag{5.10}$$

Where $P_j^0 = Wp_j$, $P_j^n = Wp_{j+1}$, and $[P_j^0, P_j^1, ..., P_j^{n-1}, P_j^n]$ are the control points of the curve that will be indicated as $\chi_j = [P_j^0, P_j^1, ..., P_j^{n-1}, P_j^n]$. The curve is completely contained in the convex hull formed by its control points.
Consider a Bézier curve of at least degree $n = 9$; given a control point P_j^0, given the control points between P_j^0 and P_{j+1}^0 as χ_j and given the control points between P_{j-1}^0 and P_j^0 as χ_{j-1}, we indicate as ρ_j the following sequence of control points:

$$\rho_j = [P_{j-1}^{\lfloor n/2 \rfloor + 1}, P_{j-1}^{\lfloor n/2 \rfloor + 2}, \cdots, P_{j-1}^n, P_j^0, \cdots, P_j^{\lfloor n/2 \rfloor - 1}, P_j^{\lfloor n/2 \rfloor}] \tag{5.11}$$

The derivatives of a Bézier curve can be obtained through the calculation of Bernstein polynomials basis of reduced order.

$$\frac{\mathrm{d}B_j(u)}{\mathrm{d}u} = n \sum_{i=0}^{n-1} b_i^{n-1} \cdot (P_j^{i+1} - P_j^i) \tag{5.12}$$

Therefore, the derivative of $B_j(u)$ is a Bézier curve of degree $n-1$ defined by the control points: $n(P_j^1 - P_k^0)$, $n(P_j^2 - P_k^1)$, ... , $n(P_j^n - P_k^{n-1})$. The k-th derivative can be derived as the following:

$$\frac{\mathrm{d}^k B_j(u)}{\mathrm{d}u^k} = \prod_{l=0}^{k} (n-l) \sum_{i=0}^{n-k} b_i^{n-k} D_{i_j}^k \tag{5.13}$$

with:

$$D_{i_j}^k = D_{i+1_j}^{k-1} - D_{i_j}^{k-1} \quad with\ D_{i_j}^0 = P_j^i \tag{5.14}$$

The combination of the control points of the original curve composes the control point of the derivatives. It follows that the derivatives of the Bézier curves are still Bézier curves that live inside the convex hulls composed of the new control

points. This is extremely useful propriety that allows to easily bound the value of the derivative. As an example let's consider a curve composed by the control

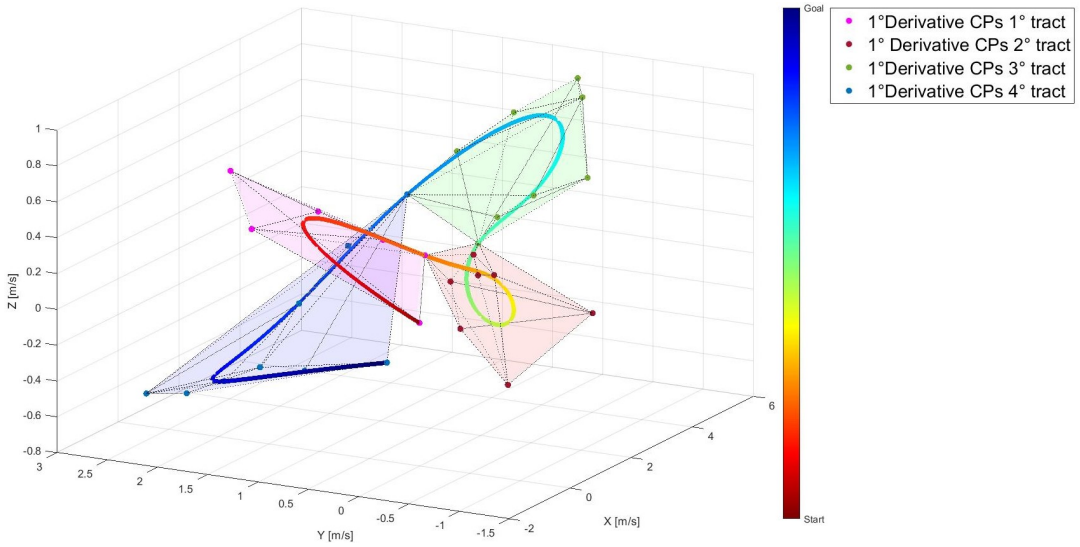


FIGURE 5.6: *An example of the first-order derivative of a trajectory composed of 4 different Bézier curves. The curves live inside the convex hull formed by the control points, allowing one to easily bound the curve.*

points $P_0, P_1, P_2, P_3, P_4, P_5$. The control points of the first derivative of the curve are $n[(P_1 - P_0), (P_2 - P_1), (P_3 - P_2), (P_4 - P_3).(P_5 - P_4)]$. Similarly, the control points of the second derivative are $n(n-1)[(P_2 - 2P_1 + P_0), (P_3 - 2P_2 + P_1), (P_4 - 2P_3 + P_2), (P_5 - 2P_4 + P_3)]$. Is interesting to notice that, due to the presence of the binomial product in the calculation of the Bézier curve, the control points of the derivative of the curve are combined following the rows of the Pascal triangle.

In order to be tracked smoothly by the quadrotor, the trajectory must be continuous and differentiable. In this paper, following the work of [38], we consider a trajectory of class C^4. Since snap is directly related to vehicle angular acceleration, and thus to the control torque acting on the quadcopter [47], this allows us to obtain trajectories that can be mapped in smooth control commands to the quadrotor. It follows that, the trajectory must be continuous in all the k^{th} derivatives, with $k \in [1, 4]$. We enforce the continuity constraints, by setting equality constraints, on control points corresponding to two consecutive curves. For the j^{th} and $(j-1)^{th}$, with $j \in [1, m-2]$ we have:

$$\left.\frac{\mathrm{d}^k B_j(t_j)}{\mathrm{d}t^k}\right|_{T_j} = \left.\frac{\mathrm{d}^k B_{j-1}(t_{j-1})}{\mathrm{d}t^k}\right|_{T_{j-1}+s_{j-1}} \tag{5.15}$$

For every condition, varying $k \in [0,4]$, we impose a linear system in P_j^i that can be expressed as follows:

$$\begin{cases} P_{j\text{-}1}^{n} = Wp_j \\ P_j^0 = Wp_j \\ P_{j\text{-}1}^{n\text{-}1} + a\left(P_j^1 - P_j^0\right) - P_j^0 = 0 \\ -P_{j\text{-}1}^{n\text{-}2} + a^2\left(P_j^2 - 2P_j^1 + P_j^0\right) - P_j^0 + 2P_{j\text{-}1}^{n\text{-}1} = 0 \\ P_{j\text{-}1}^{n\text{-}3} + a^3\left(P_j^3 - 3P_j^2 + 3P_j^1 - P_j^0\right) - P_j^0 + 3P_{j\text{-}1}^{n\text{-}1} - 3P_{j\text{-}1}^{n\text{-}2} = 0 \\ -P_{j\text{-}1}^{n\text{-}4} + a^4\left(P_j^4 - 4P_j^3 + 6P_j^2 - 4P_j^1 + P_j^0\right) - P_j^0 + 4P_{j\text{-}1}^{n\text{-}3} - 6P_{j\text{-}1}^{n\text{-}2} + 4P_{j\text{-}1}^{n\text{-}1} = 0 \end{cases} \tag{5.16}$$

Where $a = \frac{\dot{f}_j(T_j)}{\dot{f}_{j\text{-}1}(T_{j-1}+s_{j-1})}$. The system (5.16) can be written in a more compact form:

$$A_{con_j}\rho_j = b_{con_j} \tag{5.17}$$

Where:

$$A_{con_j} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & a-1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & +2 & 0 & a^2-1 & 2a^2 & -a^2 & 0 & 0 \\ 0 & 1 & -3 & +3 & 0 & a^3-1 & 3a^3 & -3a^3 & a^3 & 0 \\ -1 & 4 & -6 & +4 & 0 & a^4-1 & 4a^4 & -6a^4 & 4a^4 & -a^4 \end{bmatrix}$$

$$b_{con_j} = \begin{bmatrix} Wp_j \\ Wp_j \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Given $\gamma_l^{(k)}$ as the lower bound value of the k^{th} derivatives of the quadrotor model, and $\gamma_u^{(k)}$ as the upper bound value, we want to impose:

$$\gamma_l^{(k)} \leq \Gamma(t)^{(k)} \leq \gamma_u^{(k)}, \quad 1 \leq k \leq 4$$

Therefore, the generated trajectory is dynamically feasible. We can express the bounds on the derivatives values in the following form:

$$\begin{cases} \gamma_l^{(1)} \leq n\dot{f}_j(t_j)\left(P_j^{i+1} - P_j^i\right) \leq \gamma_u^{(1)} \\ \gamma_l^{(2)} \leq \frac{n!}{(n-2)!}(\dot{f}_j(t_j))^2\left(P_j^{i+2} - 2P_j^{i+1} + P_j^i\right) \leq \gamma_u^{(2)} \\ \gamma_l^{(3)} \leq \frac{n!}{(n-3)!}(\dot{f}_j(t_j))^3\left(P_j^{i+3} - 3P_j^{i+2} + 3P_j^{i+1} - P_j^i\right) \leq \gamma_u^{(3)} \\ \gamma_l^{(4)} \leq \frac{n!}{(n-4)!}(\dot{f}_j(t_j))^4\left(P_j^{i+4} - 4P_j^{i+3} + 6P_j^{i+2} - 4P_j^{i+1} + P_j^i\right) \leq \gamma_u^{(4)} \end{cases} \tag{5.18}$$

where $i \in [0, \lfloor \frac{n}{2} \rfloor - k]$. By considering also the bounds for the tract $(j-1)^{th}$ with $i \in [\lfloor \frac{n}{2} \rfloor + 1, n-k]$, we can express the (5.18), with j and $j-1$, in matrix form in the following way:

$$A_{dyn_j}\rho_j \leq b_{dyn_j} \tag{5.19}$$

where

$$A_{dyn_j} = \begin{bmatrix} A_{dyn_j}^{up} \\ A_{dyn_j}^{low} \end{bmatrix} \qquad b_{dyn_j} = \begin{bmatrix} b_{dyn_j}^{up} \\ b_{dyn_j}^{low} \end{bmatrix}$$

where, $A_{dyn_j}^{up} = -A_{dyn_j}^{low}$, $b_{dyn_j}^{up} = -b_{dyn_j}^{low}$ but with $\gamma_l^{(k)}$ instead of $\gamma_u^{(k)}$.

$$A_{dyn_j}^{up} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -a_1 & a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a_1 & a_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_1 & a_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -a_1 & a_1 \\ 0 & 0 & 0 & 0 & 0 & a_2 & -2a_2 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_2 & -2a_2 & a_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_2 & -2a_2 & a_2 \\ 0 & 0 & 0 & 0 & 0 & -a_3 & 3a_3 & -3a_3 & a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -a_3 & 3a_3 & -3a_3 & a_3 \\ 0 & 0 & 0 & 0 & 0 & a_4 & -4a_4 & 6a_4 & -4a_4 & a_4 \end{bmatrix}$$

$$b_{dyn_j}^{up} = \begin{bmatrix} \gamma_u^{(1)} \\ \gamma_u^{(1)} \\ \gamma_u^{(1)} \\ \gamma_u^{(1)} \\ \gamma_u^{(2)} \\ \gamma_u^{(2)} \\ \gamma_u^{(2)} \\ \gamma_u^{(3)} \\ \gamma_u^{(3)} \\ \gamma_u^{(4)} \end{bmatrix}$$

where $a_1 = n\dot{f}_j(t_j)$, $a_2 = \frac{n!}{(n-2)!}(\dot{f}_j(t_j))^2$, $a_3 = \frac{n!}{(n-3)!}(\dot{f}_j(t_j))^3$, $a_4 = \frac{n!}{(n-4)!}(\dot{f}_j(t_j))^4$. Is

worth stressing the fact that, the bounds we are considering are not the real bounds on the actuators but this approximation allows us to have simpler linear constraints in the optimization problem.
The quadrotor has to navigate through an environment with obstacles. The environment is considered mapped by a 3D mapping procedure. We suppose to have available the information of obstacle-free regions between pairs of points. We define obstacle-free regions $O_j^f \subseteq \mathbb{R}^3$ as convex polygons with $Wp_j, Wp_{j+1} \in O_j^f$. O_j^f are defined as solutions of the following set of linear inequalities (h-representation of the polygon):

$$\begin{gathered} a_j^{11}x_j^i + a_j^{12}y_j^i + a_j^{13}z_j^i \leq \mu_j^1 \\ a_j^{21}x_j^i + a_j^{22}y_j^i + a_j^{23}z_j^i \leq \mu_j^2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots \\ a_j^{\eta 1}x_j^i + a_j^{\eta 2}y_j^i + a_j^{\eta 3}z_j^i \leq \mu_j^\eta \end{gathered}$$

where η indicates the number of half-planes defining the polytope. Compacted as matrix inequality, we obtain:

$$\phi_j P_j^i \leq M_j \tag{5.20}$$

where ϕ_j is an $\eta \times 3$ matrix, P_j^i is an 3×1 control point vector whose components are x_j^i, y_j^i, z_j^i with $i \in [0, n]$. M_j is an $\eta \times 1$ vector, whose components are $\mu_j^1, \mu_j^2, \cdots, \mu_j^\eta$. The obstacle-free space O^f is defined as $O^f = \bigcup O_j^f$. Defining the trajectory as in (5.7), the obstacle avoidance is guaranteed if $\Gamma(t) \in O^f, \forall t$. Given the vector of control points ρ_j, every control points of the vector are in the obstacle-free space if:

$$\Phi_j \rho_j \leq \bar{M}_j \tag{5.21}$$

where $\Phi_j = diag(\phi_{j-1}, \cdots, \phi_{j-1}, \phi_j, \cdots, \phi_j)$ is a block diagonal matrix,
and $\bar{M}_j = [M_{j-1}, \cdots, M_{j-1}, M_j, \cdots, M_j]^T$.

Proposed Solution

The main contribution of the present work, in trajectory planning, is the development of an algorithm that is able to achieve all the goals described in Chapter 5.3.2. The novel solution is introduced by the following.
Consider a Bézier curve of at least degree $n = 9$, given a control point P_j^0 we want to optimize the position of the vector of control points:

$$\rho_j = [P_{j-1}^{\lfloor n/2 \rfloor + 1}, P_{j-1}^{\lfloor n/2 \rfloor + 2}, \cdots, P_{j-1}^n, P_j^0, \cdots, P_j^{\lfloor n/2 \rfloor - 1}, P_j^{\lfloor n/2 \rfloor}]$$

Considering the trajectory expressed in (5.7), for each waypoint, we address an optimization problem in which we optimize the position of the ρ_j control points, and we build the optimized Bézier curve using the χ_{j-1} control points. Therefore, for each waypoint, we optimize the position of the 4 control points preceding the selected waypoint and the position of the successive 4 control points. It follows that once

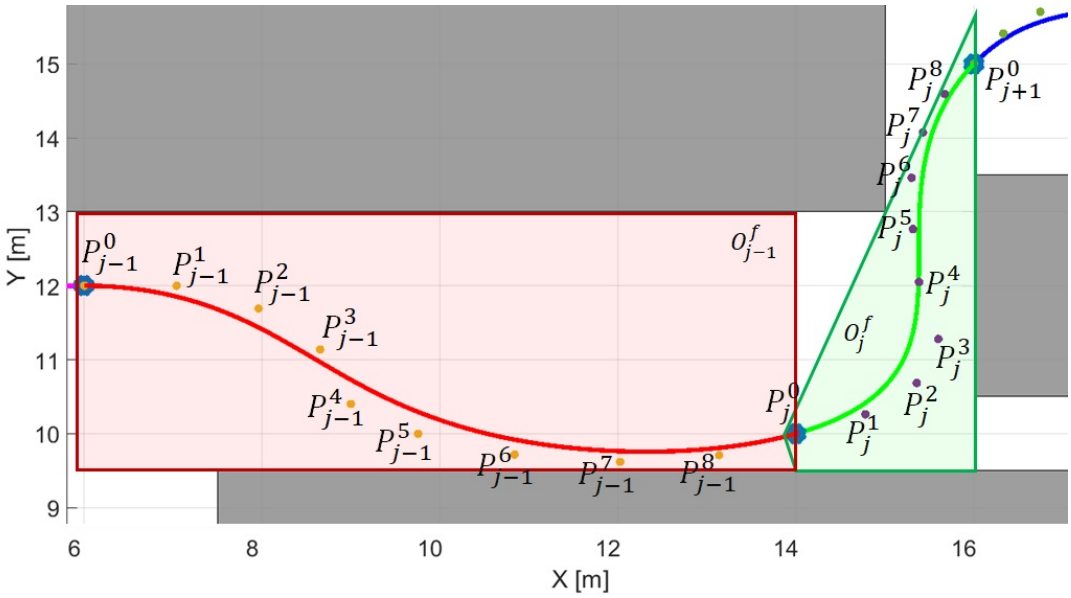


FIGURE 5.7: *Example of control points nomenclature. The collection of control points ranging from P^5_{j-1} to P^4_j is refereed as ρ_j*

obtained the optimized ρ_{j-1} and ρ_j we can build χ_{j-1} and compute the optimized Bézier curve using χ_{j-1}.

Considering m waypoints we'll end up having $m-2$ optimization problems, one for each waypoint except for Wp_0 and Wp_{m-1} because in the first and in the last tract we impose the first and last four derivatives values. Consequently, we are able to auto-determine the position of the first 4 control points and the position of the last 4 control points. In order to satisfy the dynamics and continuity constraints, the optimization problems will be solved in sequence. The optimization function we consider, for a certain $j \in [1, m-2]$, is in the following form:

$$J_j = \left(\frac{\partial^4 B_j(t)}{\partial t^4} \bigg|_{t=T_j} \right)^2 \tag{5.22}$$

The j^{th} optimization problem can be expressed in a quadratic form as in the following:

$$\begin{aligned} \min_{\rho_j} \quad & {\rho_j}^T H_j \rho_j \\ \text{s. t.} \quad & A_{con_j} \rho_j = b_{con_j} \\ & \Phi_j \rho_j \leq \bar{M}_j \\ & A_{dyn_j} \rho_j \leq b_{dyn_j} \\ & A_{new_j} \rho_j \leq b_{new_j} \end{aligned} \tag{5.23}$$

Where: A_{dyn}, b_{dyn} represent the matrices of a new set of inequality constraints.

The new framework, with the new optimization variable ρ_j and ρ_{j-1} for a tract j, imposes the usage of additional inequality constraints to ensure that the dynamic bounds are respected during the whole trajectory. This happens because we optimize over ρ_j and ρ_{j-1} but we build the trajectory over χ_{j-1}. In fact, optimizing ρ_{j-1} we fix the first half control points of the j^{th} tract, and optimizing ρ_j we fix the second half control points of the j^{th} tract, but we have no guarantee that the two sets of control points will produce a curve with bounded derivatives. In order to bound the values of the derivatives of the whole curve between Wp_{j-1} and Wp_j, we must add an additional inequality constraint. This is done in the following. Given the ρ_{j-1} as known, we can express the second half set of the control points of the tract j i.e. $[P_{j-1}^{\lfloor n/2 \rfloor+1}, P_{j-1}^{\lfloor n/2 \rfloor+2}, \cdots, P_{j-1}^{n}]$ in a way that all the derivatives, of the resulting Bézier curve, lies in a convex hull bounded by the bounds of the dynamic defined in section 5.3.2. The new constraints are:

$$\begin{cases} \gamma_l^{(1)} \leq n\dot{f}_{j-1}(t_{j-1})(D^1_{\lfloor n/2 \rfloor_{j-1}}) \leq \gamma_u^{(1)} \\ \gamma_l^{(2)} \leq \frac{n!}{(n-2)!}(\dot{f}_{j-1}(t_{j-1}))^2(D^2_{(\lfloor n/2 \rfloor+i-2)_{j-1}}) \leq \gamma_u^{(2)} \\ \gamma_l^{(3)} \leq \frac{n!}{(n-3)!}(\dot{f}_{j-1}(t_{j-1}))^3(D^3_{(\lfloor n/2 \rfloor+i-3)_{j-1}}) \leq \gamma_u^{(3)} \\ \gamma_l^{(4)} \leq \frac{n!}{(n-4)!}(\dot{f}_{j-1}(t_{j-1}))^4(D^4_{(\lfloor n/2 \rfloor+i-4)_{j-1}}) \leq \gamma_u^{(4)} \end{cases} \tag{5.24}$$

where $i \in [1, k]$. It is possible to express (5.24) in a matrix form:

$$A_{new_j}\rho_j \leq b_{new_j} \tag{5.25}$$

where $A_{new_j} = \begin{bmatrix} A^{up}_{new_j} \\ A^{low}_{new_j} \end{bmatrix}$ $b_{new_j} = \begin{bmatrix} b^{up}_{new_j} \\ b^{low}_{new_j} \end{bmatrix}$

where $A^{up}_{new_j} = -A^{low}_{new_j}$, $b^{up}_{new_j} = -b^{low}_{new_j}$ but considering $\gamma_l^{(k)}$ instead of $\gamma_u^{(k)}$.

$$A^{up}_{new_j} = \begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2a_2 & a_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3a_3 & -3a_3 & a_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3a_3 & a_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4a^4 & 6a_4 & -4a^4 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6a_4 & -4a^4 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4a^4 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$b^{up}_{dyn_j} = \begin{bmatrix} \gamma_u^{(1)} + a_1 P^4_{j-2} \\ \gamma_u^{(2)} - a_2 P^4_{j-2} \\ \gamma_u^{(2)} - a_2 P^4_{j-2} + 2a_2 P^3_{j-2} \\ \gamma_u^{(3)} + a_3 P^4_{j-2} \\ \gamma_u^{(3)} + a_3 P^4_{j-2} - 3a_3 P^3_{j-2} \\ \gamma_u^{(3)} + a_3 P^4_{j-2} - 3a_3 P^3_{j-2} + 3a_3 P^2_{j-2} \\ \gamma_u^{(4)} - a_4 P^4_{j-2} \\ \gamma_u^{(4)} - a_4 P^4_{j-2} + 4a_4 P^3_{j-2} \\ \gamma_u^{(4)} - a_4 P^4_{j-2} + 4a_4 P^3_{j-2} - 6a_4 P^2_{j-2} \\ \gamma_u^{(4)} - a_4 P^4_{j-2} + 4a_4 P^3_{j-2} - 6a_4 P^2_{j-2} + 4a_4 P^1_{j-2} \end{bmatrix}$$

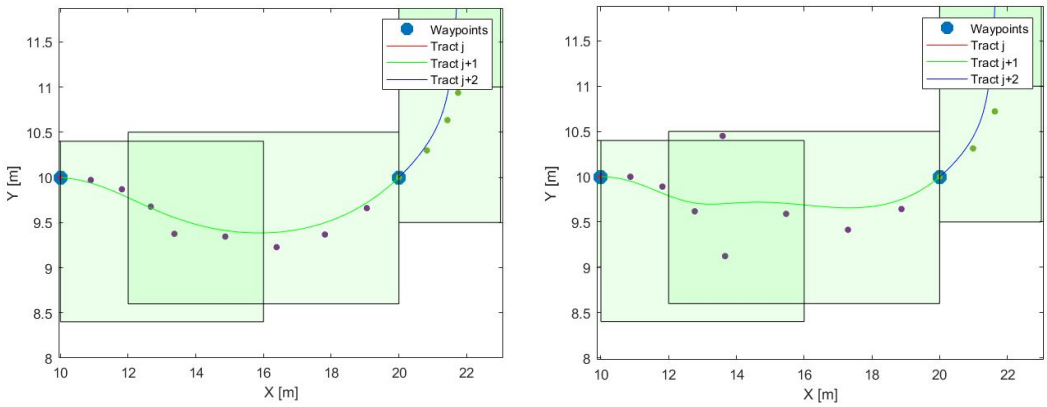
where, $a_1 = n\dot{f}_{j-1}(t_{j-1})$, $a_2 = \frac{n!}{(n-2)!}(\dot{f}_{j-1}(t_{j-1}))^2$, $a_3 = \frac{n!}{(n-3)!}(\dot{f}_{j-1}(t_{j-1}))^3$, $a_4 = \frac{n!}{(n-4)!}(\dot{f}_{j-1}(t_{j-1}))^4$.

As an example, suppose to optimize the position of the control points in ρ_j. After that, let's calculate the control points ρ_{j+1} without the additional dynamic constraints. In the end, we obtain two sets of control points:

$$\rho_j = \left[P^5_{j-1}, P^6_{j-1}, P^7_{j-1}, P^8_{j-1}, P^9_{j-1}, P^0_j, P^1_j, P^2_j, P^3_j, P^4_j\right]$$

$$\rho_{j+1} = \left[P^5_j, P^6_j, P^7_j, P^8_j, P^9_j, P^0_{j+1}, P^1_{j+1}, P^2_{j+1}, P^3_{j+1}, P^4_{j+1}\right]$$

It is now possible to build the j^{th} curve using the control points $\chi_j = [P^0_j, P^1_j, \cdots, P^8_j, P^9_j]$. Since the constraints in eq. 5.25 weren't implemented, the control points $P^5_j, P^6_j, P^7_j, P^8_j$, calculated in ρ_{j+1}, does not take into account the presence of the previous control points $P^1_j, P^2_j, P^3_j, P^4_j$, calculated in ρ_j. This could lead to unbounded derivatives of the resulting trajectory. This happens because some control points of the derivatives of the tract are not bounded. For the sake of clarity, the control points are given by $9(P_5 - P - 4)$ and $72(P_6 - 2P_5 + P_4), 72(P_5 - 2P_4 + P_3)$ and the other relatives to the third and fourth derivative are not bounded, leading to a generated trajectory that cannot be perfectly tracked by the UAV, as explained in section 5.3.1.



(A) (B)

FIGURE 5.8: *Examples of the effect on control points of the additional dynamic constraints. In* (A) *the curve in the* j^{th} *tract with the additional constraints. The curve appears smooth and the derivatives are bounded because every control point of the derivatives is bounded. In* (B) *the same curve but without the additional dynamic constraints. It is possible to see that the control points tend to displace without taking into account the presence of the control points of the previous curve. This causes a lesser smooth curve and, more importantly, the lack of control in determining the convex hull of the derivatives. Consequently, we have no insurance that the derivatives will stay bounded for the whole duration of the tract* j

The computational complexity of convex quadratic programming is $O(v^3)$ [50], where v is the number of variables of the problem. In problems in the form of the one reported in [25], the computational complexity is given by $O\left((mn)^3\right)$. Instead in (5.23) we solve $m-1$ problems with only n variables, i.e. solving an optimization problem for each piece of the trajectory the computational complexity becomes $O\left(mn^3\right)$, which means scaling down with the square of the number of trajectory pieces.

As indicated in section 5.3.1, the re-planning is intended as the ability to re-calculate some part of the curve when some input changes (i.e. the waypoints location, the addition/removal of waypoints, changes in O_j^f or changes in the time span). Based on the events triggering the re-planning the procedure to replan is the following:

- Change in Wp_j without changing time spans: solve the j^{th} problem (5.23);
- Change in time span s_j: the time span of a trajectory affects 2 optimization problems (5.23) which need to be solved for j and $j-1$;
- Change in Wp_j changing the two adjacent time span s_{j-1}, s_j: 3 optimization problems (5.23) need to be solved for $j-1, j, j+1$;
- Change in O_j^f: the free area affects 2 optimization problems (5.23) which need to be solved for $j, j+1$;

After the optimization problems are solved, the affected trajectories must be recomputed. If the optimization problem (5.23) with index j is solved, then the trajectories

with index j and $j-1$ are affected. This means that before the drone is able to fly through the j^{th} tract, the $(j+1)^{th}$ problem is solved i.e. all the control points of the j^{th} tract and half control points of the $(j+1)^{th}$ are calculated. As a general re-planning example, if two consecutive waypoints with index $j = 3, 4$ and their time spans are modified, then 4 optimization problems with $j = 2, 3, 4, 5$ need to be computed and trajectories with $j = 1, 2, 3, 4, 5$ recomputed from the newly obtained control points.

Differences with Off-The-Shelf Local Approach

The proposed solution allows adopting a local approach in solving the planning problem that guarantees the respect of the proprieties in section 5.3.2 even if the trajectory is modified. As an example, consider a trajectory planning problem to be solved using the Bézier curve to guarantee the proprieties in section 5.3.2 between Wp_0 and Wp_{m-1} passing through m waypoints. Suppose a time allocation for each tract. A global approach could solve the problem and guarantee all the wanted proprieties, but we need to know all the waypoints in advance and we have to recompute the whole trajectory in case a waypoint or a time span changes. In the situation in which the map is not fully known is not suitable to use a global approach, as explained in section 5.3.1. We could use an off-the-shelf local approach with Bézier curves. In doing so we can create a piecewise trajectory by considering a limited amount of waypoints at times. Suppose to calculate the trajectory until waypoint Wp_j. This fixes the position of the control points $\chi_{j-1} = [P^0_{j-1}, P^1_{j-1}, \cdots, P^8_{j-1}, P^9_{j-1}]$. In calculating the next tract we want to impose the continuity until the fourth derivative. This imposes equality constraints that subtract degrees of freedom from the problem. As a consequence, the positions of the control points $[P^0_j, P^1_j, P^2_j, P^3_j, P^4_j]$ are fixed and may lie outside O^f_j leading to a possible collision with obstacles. It is clear that using the off-the-shelf local approach, the more degrees of continuity we want to impose, the more is the risk to lie outside the safe areas. This problem is present during a sequential creation of the curves and if some curve has to be modified. Instead, using the proposed approach, where we optimize over ρ_j rather than χ_j, we are able to guarantee the respect of all the proprieties described in section 5.3.2. This is true during the sequential creation of the trajectory and in case of changes in waypoint location and/or time spans.

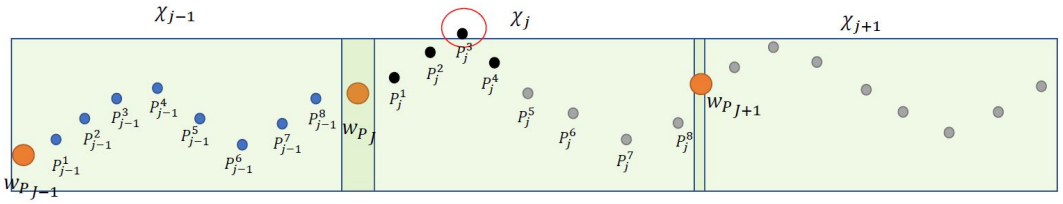


FIGURE 5.9: *Example of the application of the off-the-shelf approach. Due to continuity constraints, the collection of control points $[P_{j-1}^5, P_{j-1}^6, P_{j-1}^7, P_{j-1}^8, P_{j-1}^9]$ fixes the position of the first control points of the next tract, and might forces some control points to lie outside the safe area.*

Results and Conclusions

In order to show the results of the algorithm, we implemented a simulation in a map with 5 convex obstacles with waypoints placed at $Wp_1 = [0, 6, 0], Wp_2 = [6, 12, 1.4], Wp_3 = [14, 10, 0 - 4], Wp_4 = [16, 15, 0.6], Wp_5 = [23, 16, 1.4]$ with a time span vector of $[5s, 4s, 4s, 3.5s]$. The position of the obstacles, and the free spaces, are considered known. The trajectory needs to fulfil all the proprieties expressed in section 5.3.2. The algorithm is applied from Wp_1 to Wp_5 and takes into account the possibility of modifying one waypoint and/or a component of the time span vector. The resulting trajectory can be seen in fig.5.11 and in fig. 5.12, the resulting derivatives trajectories can be seen in fig. 5.10.

The algorithm developed allows us to create optimal trajectories that are functionally controllable, for quadrotors, and that guarantee a safe path to avoid obstacles. The main advantage of the developed method is the possibility to perform local changes of the trajectory, without the need to re-compute the whole curve. This makes possible a local re-computation of the trajectory if unplanned obstacles pop up or if the map changes, making the algorithm suitable to be applied when there is no a priori knowledge of the map.

Additionally, it should be possible to boost the execution time by parallelizing the optimization problems by considering a set of optimization problems from the initial waypoint until some middle waypoint, and in parallel address another set of optimizations from the final waypoint to the middle.

In future works, we'll aim to implement the algorithm in Gazebo/ROS, releasing the package as a free repository, and implementing it on the AlCan02 UAV.

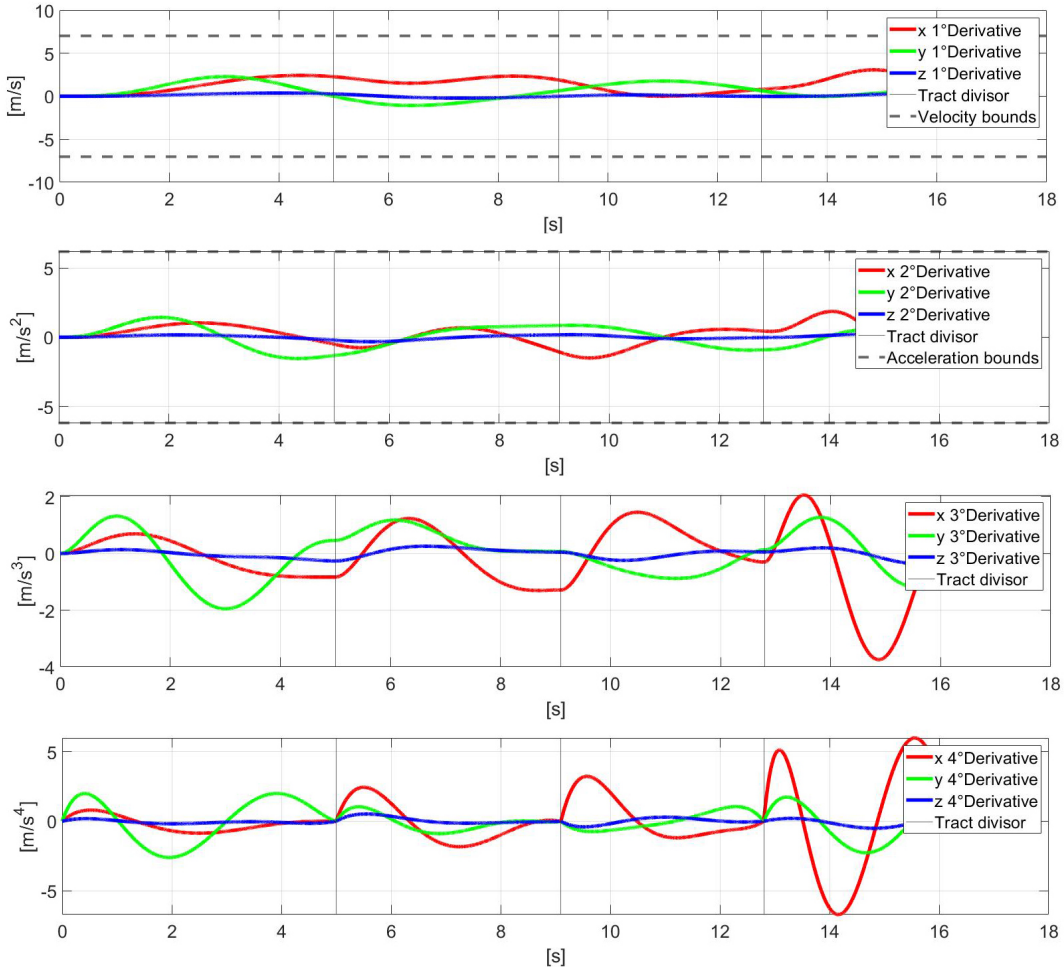


FIGURE 5.10: *The resulting derivatives trajectories of the simulation using the proposed kinodynamic optimal trajectory planner with Bézier curves. The continuity is guaranteed until the snap, as can be seen from the figure. The derivatives values are bounded in order to be feasible with the vehicle.*

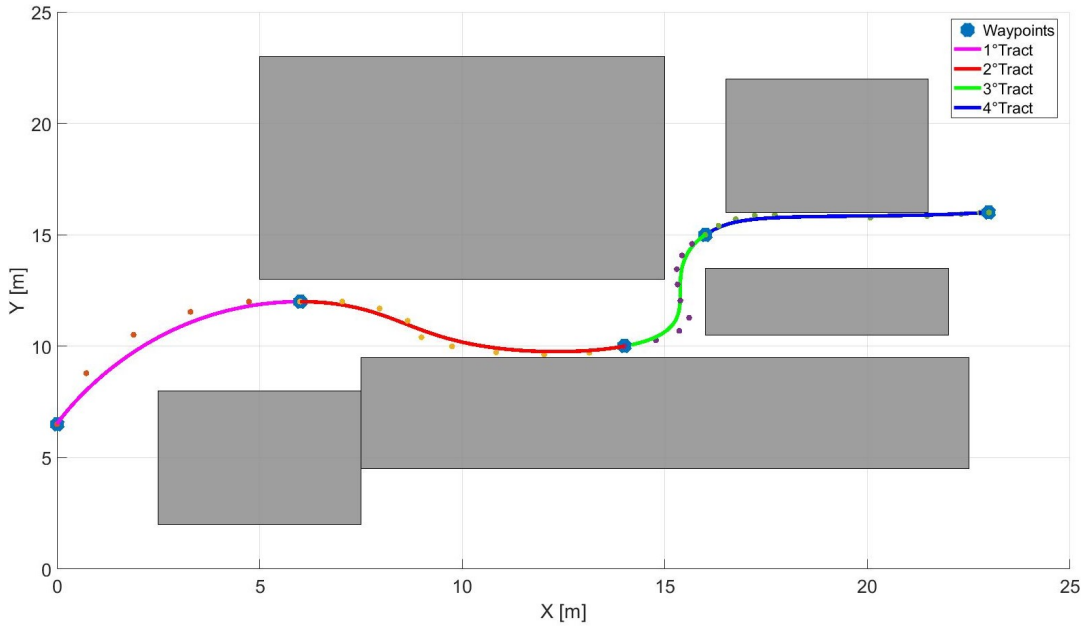


FIGURE 5.11: *The resulting trajectory of the simulation test using the proposed kinodynamic optimal trajectory planner with Bézier curves. The snap continuity allows a smooth geometric path.*

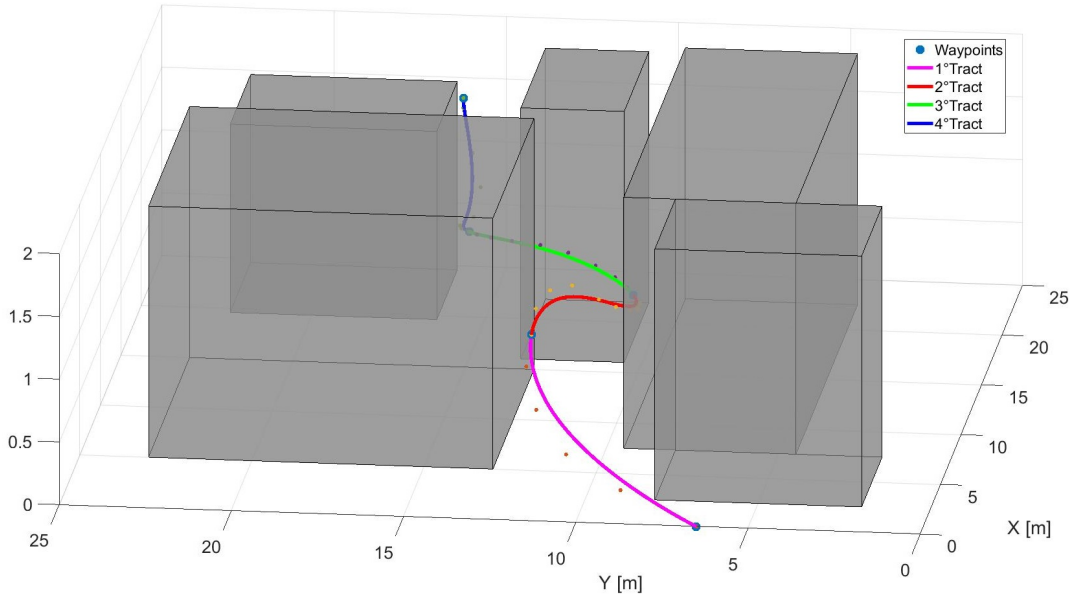


FIGURE 5.12: *The 3D view of the resulting trajectory of the simulation test using the proposed kinodynamic optimal trajectory planner with Bézier curves.*

6 Experiments and Real Scenarios Applications

The design, prototyping, and construction of the Alcan02 has been possible due to the huge amount of experiments on the test bench, and flight tests. While the internal tests have been made by Aslatech in order to ensure the functionalities of the machine, the field tests have been carried out by CAI. In fact, the UAV has been tested in typical real SAR scenarios, in order to validate and evaluate the quadrotor performance. This chapter will present the methods used to test the Alcan02 and the experiments and real scenario application carried out by CAI. In doing so will be reported the information extracted from the flight journal and logs about the test procedures, fixes, and sometimes even crashes.

6.1 Internal tests

The tests procedures to validate the compatibility with the payloads, which can be found in section 2, and the tests carried out to select the power train components, discussed in section 3, are not the only internal tests done on the Alcan02. In fact, while the discussed procedures were needed during the design phase, after the development of the Alcan02 a series of experiments have been done in order to validate the new structure.
All the components of the avionics are tested separately with a HITL procedure. As an example, let's suppose to have the board A to be testes. The board has to be connected to board B and C. A test firmware is usually created, simulating the signals that might come from boards B and C. This procedure is repeated until all the components of the avionics are validated. A particular focus has been put on testing the electronics of the autopilot module and the internal logic module. The two modules, after being validated individually, are connected and tested. This procedure is needed to validate all the physical connections, the signal routing and handling, and the internal logic behaviour. The connections with the motor groups are validated for each group. The PWM commands are delivered to the motor groups from the RC through the radio module and the autopilot. The full quadrotor is then assembled and tested on a test bench before going to flight. The test bench is composed of a PVC structure that houses 4 hooks tightened by a screw to keep the drone anchored when performing tests with propellers. The UAV is equipped with propellers and anchored to the test bench, so is possible to check all the functions without takeoff

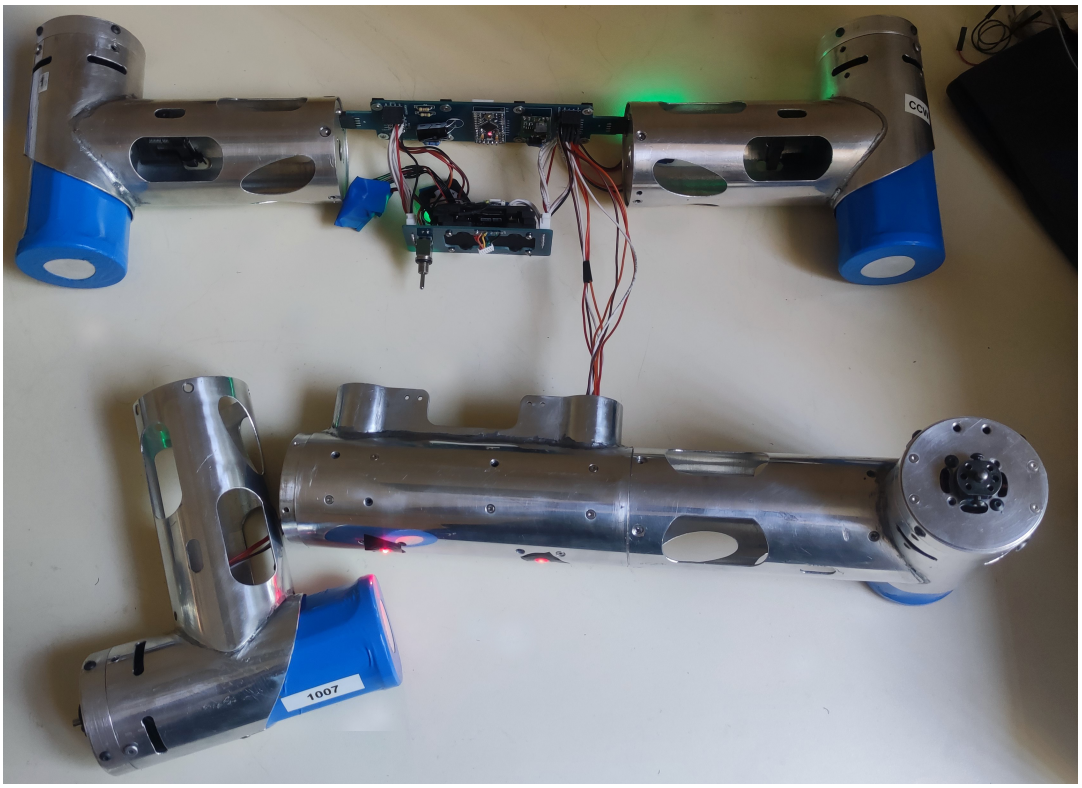


FIGURE 6.1: *Example of a testing phase of the interconnection of the two avionics modules of the AlCan02. In this particular test, the Mega2560 within the CANSX was receiving data from the autopilot within the autopilot module. The purpose of the test was to verify the chain of connections from the autopilot, to the mega2560, to the pro mini, and then to the radio module.*

the machine. This procedure helps to spot unwanted behaviour that may happen due to some failure in the battery pack or some problems that might arise in presence of great current consumption.

As an example, at the test bench, we noticed a not stable telemetry signal and a slow response to RC commands. In cases like this, it is of paramount importance to investigate the various causes of the problems with reliable and repeatable tests. Besides, the problems have to be investigated following a predefined order with the same variables or adding one variable at a time, in order to obtain comparable results. Firstly, we decided to investigate the problem of the slow response to RC command. We connected the PPM signal to an oscilloscope as in fig. 6.2, in order to check if the problem was the signal delivered to the autopilot. The PPM signal was highly disturbed only when the motors turned with propellers. Further investigations highlighted the fact that a floating ground was interfering with the Mega2560 that processes the PPM data and sends it to the autopilot. Additionally, this caused corruption of some packages of the telemetry serial protocol, leading to unstable telemetry data visualized on the RC.

FIGURE 6.2: *Setup for the test with the PPM signal delivered to the autopilot connected to an oscilloscope in order to check the integrity of the signal. The test has been repeated with and without propellers.*

Once obtained a reliable machine is at the test bench, the UAV is ready for some internal flight tests in order to check the flight performance. We developed an internal procedure to be done to verify the main functions of the machine. Once being connected with the CGS and verified the status from the telemetry, we proceed to the take-off in manual mode. The manual flight mode allows the pilot to fly the machine without position control. In fact, only the attitude controllers and rate controllers are used, leaving to the pilot the stabilization in case of external disturbances, and allowing the tune of the low-level PID controllers. Consequently, the position mode is enabled, stabilizing the position of the UAV using the GPS data and the position PID controller. Based on the wanted performance is possible to tune the position controller after having tuned the low-level controllers. The test routine to check the safety functions is then executed. In particular, auto-landing and RTL are configured based on the future operations' necessity. This is a very critical task because a wrong configuration of the safety parameters can lead to crashes or problems. As an example, in wooded environments operations, it is advisable to set the RTL height at an altitude that allows the drone to perform the returning manoeuvre over the top of the trees. Moreover, in alpine environments with strong wind gusts, it is advisable to prefer autolanding or set the RTL with enough battery remaining. This is justified by the fact that, with empty batteries, in order to compensate for the wind, the UAV has to increase the PWM inputs to the motors. At high quotes and with strong winds is not difficult to reach the saturation of the motors, and consequently to have the drone carried away by the wind action.

The UAV is now ready to be used in supervised flights. In order to fulfil all the

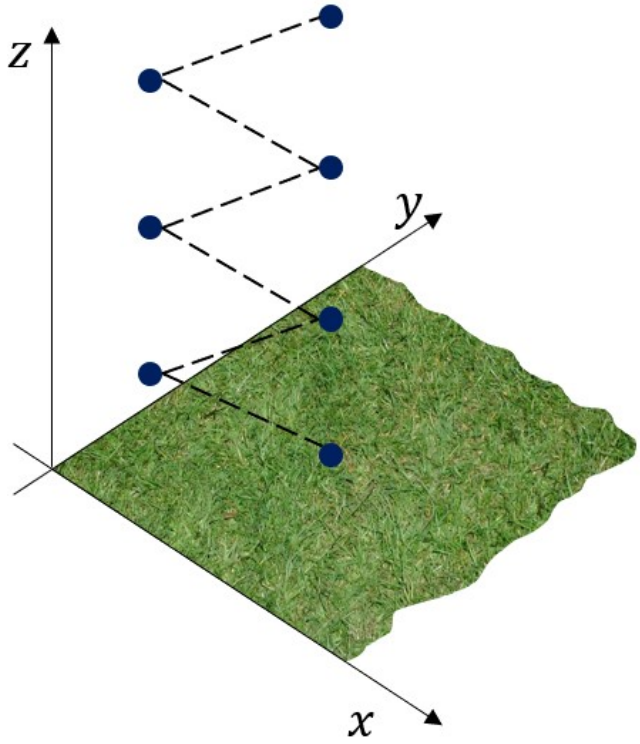


FIGURE 6.3: *Setup for the test with the PPM signal delivered to the autopilot connected to an oscilloscope in order to check the integrity of the signal. The test has been repeated with and without propellers.*

specifications of the Airborne project and deliver the UAV to the CAI for the intensive tests on real scenarios, also the automated flight have to be validated. All the types of grids are executed with different parameters and test missions are executed from the CGS. A typical mission is the execution of an inclined straight path back and forward with increasing altitude. This aims to stress the motors and check if the performance is maintained even at low battery levels and if the temperatures do not grow excessively. It is worth stressing the fact that all the supervised and automated flights are repeated with the two payloads and the reading of the payloads constantly monitored. once the performance in mission mode achieves the intended results, the UAV can be delivered to the CAI for operation in near-real and real scenarios.

6.2 Field Tests

Airborne's drone is a peculiar drone without a camera but equipped with ARTVA and RECCO receivers for specialized search and rescue missions in avalanche scenarios. Since the start of the project, the total days of flight in mountains were 352 with over 1000 flights and 557 hours dedicated to testing and flight in alpine environments. The flights were done at 70% with ARTVA payload and the remaining with RECCO antenna since RECCO antenna was completed and released later in

the project.
All the field test operations have been done using the same start-up procedure. From the arrival to the flight, only 2 minutes are assigned for the setup. In the two minutes the procedure is the following:

1. Mount the Alcan02 frame;
2. Insert the batteries;
3. Power up the RC;
4. Check the batteries status and the telemetry;
5. Arming the motors;
6. Check payload status;
7. Check number of satellites. For safety wait to have at least 10 satellites for automatic flights;
8. Takeoff and positioning on the first waypoint of the grid;
9. Determine the length of the grid;
10. Start the automatic grid search;
11. If a good signal is detected, lower the gain of the antenna (only for RECCO) and perform a fine search in position mode;
12. Alert the ground rescuers;
13. Re-iterate from point 10.

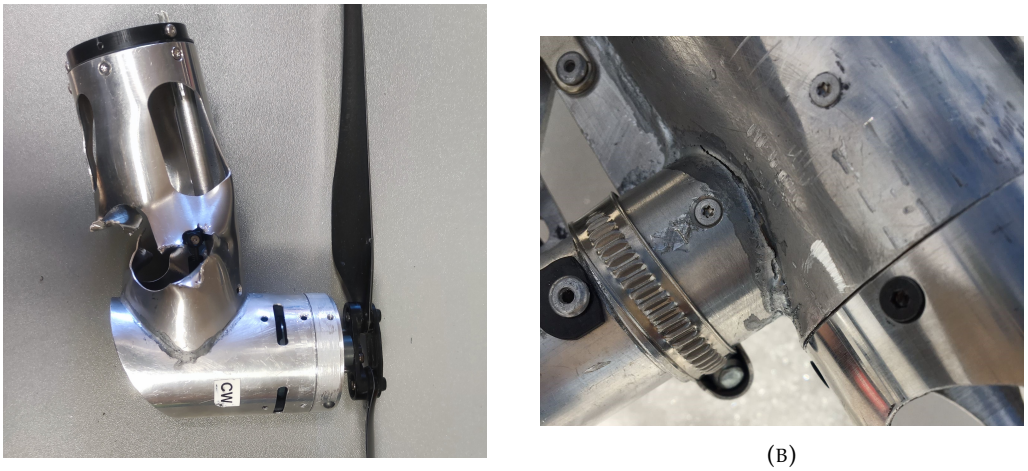
During the field tests, the pilot has no prior knowledge of the position of the transmitting devices. In fact, the ARVA beacon and the RECCO reflector are buried within search areas in positions unknown to the pilot in order to not bias the search operation.
All the data and results of the test that are presented in this chapter are extracted from the flight logs and from the flight journal written by CAI. A major part of the tests has been conducted in Rhêmes-Notre-Dame, at Saint Pierre Rumiod, ski resorts on Mont Blanc, Champorcher, Saint-Nicolas and neighbouring areas. For the sake of simplicity, the results are presented in two major sections. The first section resumes the results of the tests conducted to validate the design and capabilities of the AlCan02 quadrotor. The second section presents the results of tests conducted to validate the performances of the payloads and to improve the search strategies.

6.2.1 UAV Validation

The stress to which the drone is subject at high quotes and low temperatures is much more consistent if compared with the internal flight tests executed during the first validation of the machine. In fact, the repeated and continuous usage in harsh environmental conditions can help to spot problems that might be highlighted only during continuative usage. The tests aimed to validate the frame structure, the battery packs, and consequently the flight performances.

Frame Validation

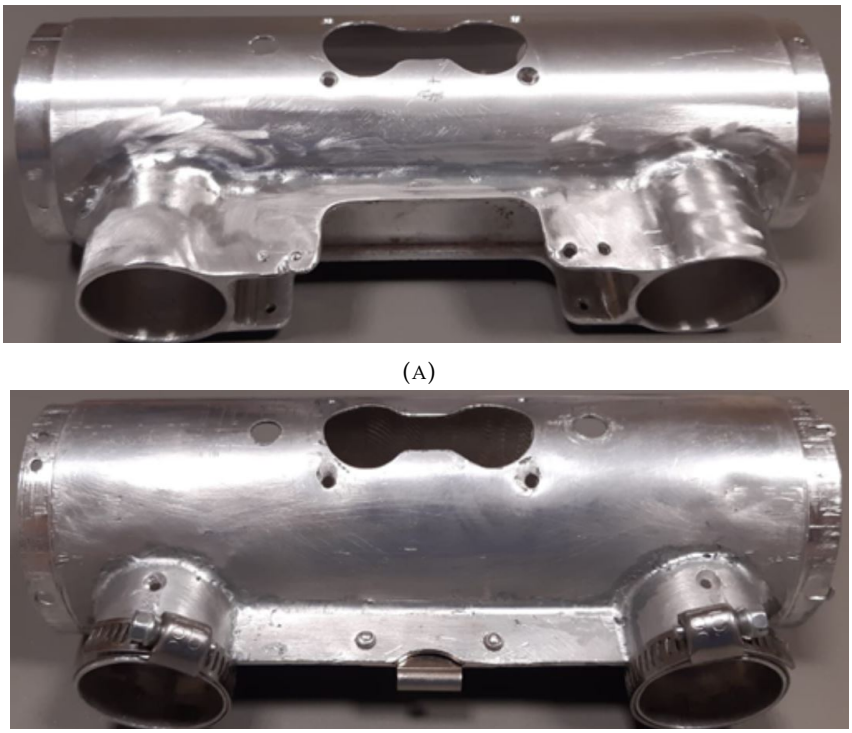
The first test session with the AlCan02 was performed at over 1700 m altitude in similar post-avalanche scenarios. Weather conditions included strong winds, snow, and sub-zero temperatures. The drone frame tended to misalign after severe stress due to wind compensation and strong motors activity. This caused an imbalance in the consumption of the motors, as some were tilted due to the frame imbalance. This required a careful inspection, and eventually a manual fix to the frame, before going to flight. It is clear that an imbalance in the activity of the motors causes an imbalance in the use of the batteries. During a long flight in late December 2021, during an RTL manoeuvre with low batteries, a battery was completely discharged, causing a crash on frozen ground that damaged the chassis.



(A) (B)

FIGURE 6.4: *The images show the damaged frame after the crash in late December 2021. In* (A) *the motor group arm. In* (B) *the deformed frame connecting the CANSX to the central module.*

The imbalance of the drone's frame caused also compensation problems around the Yaw axis in presence of strong winds. During flights with gusty wind-speed around $35Km/h$ the quadrotor presents difficulty in maintaining the Yaw angle while compensating for the wind action. In order to solve the problems, the mechanical connectors of the lateral modules with the central module have been modified to improve torsional rigidity. A new aluminium brazing system has been used to build the connectors parts by a single aluminium block. Besides, the rigidity is further



(A)

(B)

FIGURE 6.5: *The two different mechanical connectors that allows to connect the central module with the lateral modules. In* (A) *the new aluminium brazed connectors. The new structure improves the torsional rigidity in order to solve the problem highlighted during the CAI tests. In* (B) *the old version of the connector that causes misalignment in the drone frame in high-stress conditions.*

improved by a new hooking mechanism that allows to rigidly secure the connection between the central module and lateral modules.

The modifications of the frame proved to be efficient in the next flight tests. In January 2021 the UAV has been tested with temperatures around $-10°C$ with strong winds. In particular, during the last week of January, the machine has been tested, recording good performance, on terrain with $600m$ of slope and winds with gusts of $38Km/h$. Attention must be drawn to the fact that in flights with low temperatures and high humidity, ice tends to form on the frame and on the propellers. The pilot has to take care of cleaning the machine while changing the batteries before flying again.

The modularity of the frame allows the pilot to easily carry the UAV with payloads, batteries, and RC inside a backpack. This represents a key factor when operating in areas that necessitate the helicopter or cable car to be reached.

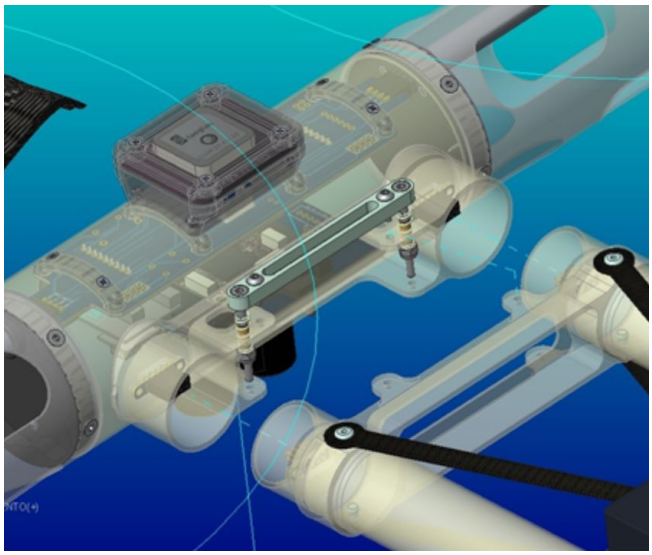


FIGURE 6.6: *The new hooking mechanism allows securing the connection between the lateral modules and the central module.*

Battery Packs Validation

The batteries quick replacements system is of critical importance during operations in real scenarios. In fact, once reached the area in which start the operation, the quick deployment of the machine depends also on how quickly the batteries can be inserted. The operation has to be comfortable to be done with gloves and with snow. The tests done by CAI showed that the system of battery replacement of the AlCan01 was not very comfortable when carried out with snow. Instead, the batteries replacement system of the AlCan02 proved to be more efficient and comfortable. The flights carried out at low temperature show that the performance of the cells Sony US18650VTC were maintained even with no pre-warm. Nonetheless, in order to avoid possible problems, if the batteries are not pre-warmed is better to not stress the machine during the first minute of flight.

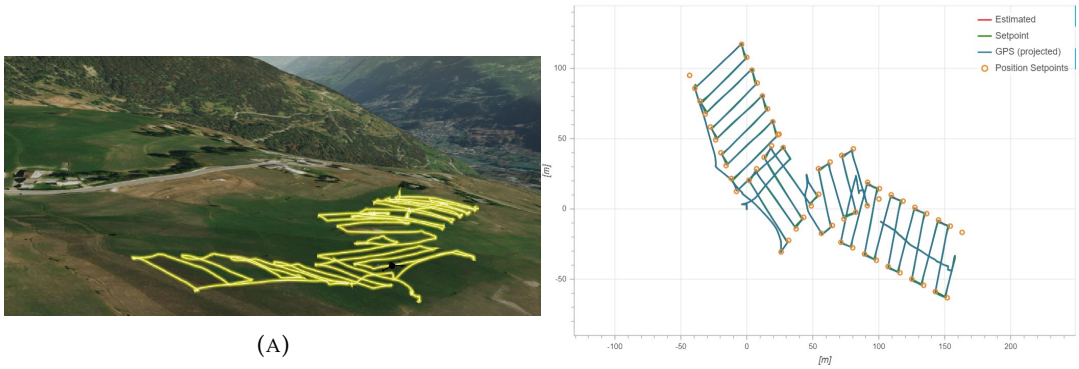
6.2.2 Payloads Validation

The integration of the ARVA antenna and RECCO antenna as payloads was the core of the Airborne project. As explained in 2.1.1 and 2.1.2, the two payloads has been produced in order to be integrated with the AlCan02. The tests on the field conducted by CAI have been a decisive factor in validating the integration and performance of the payload. The devices to be searched have been hidden in the delimited search area in order to not bias the pilot during the research.

ARVA Validation

The ARVA payload was the first to be fully tested and validated. The tests have been conducted since the implementation of the AlCan01 and have been repeated with the Alcan02. In a typical ARVA search scenario, at least one transmitting beacon

is used. The first tests highlighted the presence of false-positive detections, which might be misleading in determining the true position of the beacon. Accordingly, the ARVA detection algorithm described in section 4.2.2 has been tuned in order to reduce the presence of false detections. In the successive tests, the pilot was always able to determine the position of the beacon with the precision of a few meters. The flight has been performed both in plain and highly sloped terrain. The performance of the quadrotor in flight is strictly related to the ability to identify the signal. In fact, during a flight test in Rhêmes-Notre-Dame over a highly sloped terrain, an ARVA beacon was missed due to the misalignment of the frame (explained in section6.2.1). In detail, it caused difficulties in tracking the Yaw reference and difficulties to maintain the altitude from the terrain. As a matter of fact, during the descent of the sloped terrain, the altitude was too high and the ARVA signal was not received by the antenna. Maintaining the predetermined height from the ground would have led to signal detection.
As explained in section 4.2.2, the detection of the beacon signal is possible only after a certain acquisition time. The tuning of the parameters of the onboard recognition algorithm depends also on the translation velocity of the UAV. The pilot must keep in mind that detection in the RC during the execution of the automated grid pattern means that the signals were detected around 2s before. According to the CAI reports this did not represent a problem in founding the position of the beacons. The detection of the ARVA antenna has been validated in a radius of 20m at an altitude of less than 10m from the terrain.



(A) (B)

FIGURE 6.7: *The images show the damaged frame after the crash in late December 2021. In* (A) *the motor group arm. In* (B) *the deformed frame connecting the CANSX to the central module.*

RECCO Validation

The RECCO payload was the first to be validated since it was developed after the ARVA one. During the tests, the reflectors have been hidden in the search areas in order to not bias the research operation. Differently from the ARVA, with the RECCO payload is possible to modify the emission gain from the RC. This allows performing

a fine search manoeuvre by lowering the gain when approaching the reflectors. The search manoeuvre is usually started by using the automatic search pattern described in section 5.1 with gain 6. The maximum gain possible is 8, but with internal tests, we found that the emission power 6 is the best compromise between the dimension of the detection radius and the presence of interference. This choice has been validated by CAI tests. Once a signal is detected by the payload, is possible to stop the automated flight pattern and proceed to a manual fine search flight by lowering the gain to spot with more precision the location of the reflectors. As confirmed by the flight logs of the CAI, this is not always the best choice. In fact, it might be best to signal the position of the detection to the ground rescue team and keep searching for other reflectors. The test held on 22-02-2021 has clearly proven the utility of the AlCan02. The test has been performed in a real post avalanche scenario, with very irregular and sloped terrain. Several RECCO reflectors have been positioned in the avalanche area. The UAV flight was intensive due to the continued correction to follow the terrain. Nonetheless, the flight performance was good. The reflector has been founded with a good precision that would have allowed the ground rescue team to save the buried victims. The test highlighted the efficiency of a triangular grid pattern in a situation with such difficult terrain. In those situations, the advantage of having a flying detecting device is very clear. In fig. 2.20a the AlCan02 equipped with the RECCO payload can be seen during the test held on 22-02-2021.
The reflectors used are: ski touring boot shoes with RECCO device inserted, RECCO white classic plate, RECCO clothing strip and RECCO plate on the Ortovox beacon. Flight and ground tests showed that all the types of reflectors respond in the same way, except for the Ortovox plate. In fact, this kind of plate is more difficult to be detected.
As a typical application example, we report a flight test that has been carried out on the final part of a pear-shaped avalanche with an undulating slope. The reflector has been placed in the area traced on a map and reported on Garmin with an accuracy of $3m$. The reflectors are not buried but laid on the ground. The pilot had no vision or knowledge of the position of the reflectors. During the flights, a series of triangular grids have been used during the first uphill manoeuvres to detect the first detection positions. The downhill manoeuvres have been performed with the purpose to confirm the positions of the detections. Once detected and confirmed the position of two reflectors the downhill grid has been repeated in search of a third reflector without success. At the end of the flight, the team communicated to the pilot that there were just two reflectors inside the area. The full trajectory and the position of the two reflectors can be seen in fig. 6.10. The flight has been operated at an altitude from the terrain of $18m$ and a line-to-line distance of the grid of $14m$. Particular attention must be put to the fact that the presence of the wind, in the grid direction, can afflict the detections of the antenna. In fact, the drone tends to tilt more towards the wind to counter its strength. Since the antenna is rigidly attached to the drone frame it also tilts, moving the detection lobe. As a result, the detection

FIGURE 6.8: *The uphill scan where the two reflectors were initially detected.*

points are biased. In those cases, it is the pilot and rescuers to better interpret situations with a certain wind direction by moving the search in opposite direction. In presence of husting wind, the scenario is more delicate and it might need to fly over the same area with different angles in order to identify with a certain precision the position of the reflectors. It is worth stressing the fact that, chasing perfection in the detection might be useful during training and test flights but in real operations, it is not advisable. In fact, is more efficient to signal the position of the detection to the ground team immediately if in a radius of $10m$.
The RECCO payload has been validated with a detection radius of $22m$ with a gain of 6 and an altitude from terrain between $15m$ and $20m$.

FIGURE 6.9: *The downhill scan where the positions of the two reflectors were identified and confirmed. In the image, the reflectors positions are indicated in the yellow circles.*

FIGURE 6.10: *The whole search path was executed to scan the area, in white, in order to find the two reflectors.*

7 Conclusions and Future Works

In this thesis we presented the quadrotor developed during the Airborne project. In particular, we described the methods used to design the frame structure and the modular approach that allows the UAV to be integrated with the payloads RECCO and ARVA. Moreover, we discussed the avionics and the power train groups, describing the design process and the motivation behind certain choices. The signal routing and data handling has been described, reporting the algorithms present onboard and presenting an ARVA signal detection algorithm that uses a regressive convolutional neural network to estimate the position of a transmitting beacon. A part of this work has been dedicated to discussing the problem of navigation and path planning in SAR application and has been presented the navigation algorithm implemented in the Alcan02. Moreover, we described a novel trajectory planner that uses high degree Bézier curves to obtain optimal, safe, and dynamically feasible trajectories that can be replanned locally. Finally, in order to validate the developed quadrotor, we described the tests that have been done by us to test the machine and the procedure performed by the Airborne partner CAI to evaluate the performance of the quadrotor in field applications.

The overall result of this work is a modular quadrotor constituted of an aluminium alloy shield that allows the integration of EMI sensible payloads. The shielding material can be changed accordingly to the kind of EMI we expect. Consequently is possible to extend the concept behind the development of this machine to other quadrotors to be applied in different contexts.

Modularity presents different advantages. In the first place, it allows the replacement of the 3 modules and makes it possible to obtain a quadrotor that can be easily dismounted and carried inside a backpack. It is trivial that the design allows also an easy mounting procedure. Moreover, the integration with EMI sensible payloads is furthermore increased by the decentralization of the avionics, which is moved from the centre of the quadrotor to the arms. Finally, the motor group management and the battery replacements are handled comfortably and possible in an open field situation where there are no flat surfaces to be used as a table. Besides, the mechanical innovations, the Alcan02 presents also a data handling system that allows the user to quickly monitor the state of the drone through the RC. Moreover, the data are stored to be analyzed offline. This allows complete supervision of the performance of the machine. The developed and implemented algorithms allow the complete integration with payloads, delivering the information concerning detection of the victims directly to the pilot through the RC. Furthermore, the innovations in the navigation

algorithms allow a quick-deployable and efficient automatic flight pattern that can be modified in real time by the user.
The performance, and features, of the Alcan02, proved that the quadrotor can really help in accomplishing SAR operations safely and quickly. In fact, the integration with two of the major SAR technologies (i.e. RECCO and ARVA) is a key factor in successfully achieving this task. The final machine presents a flight autonomy of around 25 minutes, an ARVA detection radius of 20m, a RECCO detection radius of 22m, a complete telemetry check system, an automatic grid pattern algorithm, a quick battery replacement system, a modular frame that allows a quick dismount and quick deployment, and the fulfilling of all of the Airborne's requirements.

7.1 Future Works

Despite the achievement of the Airborne's requirements and the development of a TRL8 machine, some topics remain still open:

- Implementation of ARVA neural network onboard: The performance will greatly benefit from the integration of the algorithm described in section 4.2.3 A future step in improving the machine is the onboard implementation of such an algorithm;

- Implementation of the trajectory generator onboard: The novel trajectory generator presented in section 5.3.2 will be integrated with the PX4 autopilot in a ROS/Gazebo simulator, then it will be integrated within the real AlCan02;

- Legacy of the Airborne project: The meaning of the Airborne project is not only the development of a single SAR quadrotor but the creation of technological seeds which will constitute a European service network aimed to boost SAR operations with the application of Alcan02 quadrotors. We, and the majority of the project partners, are still moving towards this goal, even if the project is officially ended in 2021. The next step is the production of more AlCan02 quadrotors to be given to the major alpine centres. Then, the tech seeds will be created. Even if we have now the technology to accomplish the task it still remains very challenging. Nonetheless, we are motivated to reach the goal of creating an infrastructure that can help to improve SAR operations and consequently can help to save lives.

A Quadrotor Model

A mathematical model of a quadrotor can be derived using the Newton-Euler equations of motion of a rigid body in space configuration $\mathbb{R}^3$. By assuming that the body frame has its axis aligned with the principal axis of inertia of the rigid body and considering $F_c = \{O_c, i_c, j_c, k_c\}$ as the inertial coordinate frame, we can express the dynamical model of the quadrotor, with respect to the inertial frame, as:

$$\begin{aligned} M_m \ddot{p} &= R_\psi f_c + M_m g e_3 + f_d \\ J_m \dot{\omega} &= -\omega \times J_m \omega + \tau_c + \tau_d \end{aligned} \tag{A.1}$$

where, M_m is the total mass of the quadrotor, J_m is his inertia, $p = [x, y, z]^T$ is the position of the center of mass, ω the angular velocity in body frame, and finally e_3 is the unit vector $e_3 = [0, 0, 1]^T$. The resulting force and torque generated by disturbances are represented, respectively, by f_d and τ_d. We indicate the control force and control torque vector as $f_c \in \mathbb{R}^3$ and $\tau_c \in \mathbb{R}^3$ applied to the center of mass. By considering the individual masses of the power train groups with M_i with inertia $J_{A,i}$ with $i \in [1, 4]$ expressed in their reference frame, the mass of the single payload module M_P with inertia J_P expressed in its reference frame, and neglecting the mass of the link connecting the frame to the power train groups, we can write the total inertia matrix as:

$$J_m = \sum_{i=1}^{4} R_{\psi_i} J_{A,i} R_{\psi_i}^T + \sum_{i=1}^{4} M_i J_i(r_i, h_i, \psi_i) + R_{\psi_i} J_P R_{\psi_i}^T + M_P J_5(r_5, h_5, \psi_5) \tag{A.2}$$

where $J_k(r_k, h_k, \psi_k) = \begin{bmatrix} r_k^2 \sin^2 \psi_k + h_k^2 & 0 & 0 \\ 0 & r_k^2 \cos^2 \psi_k + h_k^2 & 0 \\ 0 & 0 & r_k^2 \end{bmatrix}$

$k \in [1, 5]$, r_k with $k \in [1, 4]$ is the horizontal distance between a motor group and the centre of mass of the quadrotor, r_5 is the distance between the payload position and the centre of mass of the quadrotor, h_k with $k \in [1, 4]$ is the vertical distance between each motor group and the centre of mass of the quadrotor, h_5 is the vertical distance between the payload and the centre of mass of the quadrotor, ψ_k with $k \in [1, 4]$ is the angle required to align the frame of the single motor groups with the body frame of the quadrotor F_b, ψ_5 is the angle required to align the frame of the payload with the body frame of the quadrotor F_b.

The total mass matrix can be expressed as:

$$M_m = \sum_{i=1}^{4} M_i + M_P \tag{A.3}$$

We can indicate R_{ψ_i} as the rotation matrix relating a particular reference frame and body frame, as follows:

$$R_{\psi_i} := \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The motors' thrust is the driving force of the quadrotor and can be calculated, for a single motor group with $i \in [1, 4]$, as follows:

$$T_i = c_{T_i} \rho A_r r^2 \omega_i^2 \tag{A.4}$$

where c_{T_i} is the thrust coefficient for a specific rotor, ρ is the air density, A_r is the section area of the propeller's rotation, r is the rotor's radius, ω is the angular velocity of the rotor. The relation can be simplified in the following:

$$T_i = C_T \omega_i^2 \tag{A.5}$$

where C_T is the lumped parameter thrust coefficient obtained considering the same thrust coefficient for all the motors. Similarly, we can express the lumped torque relation with the following:

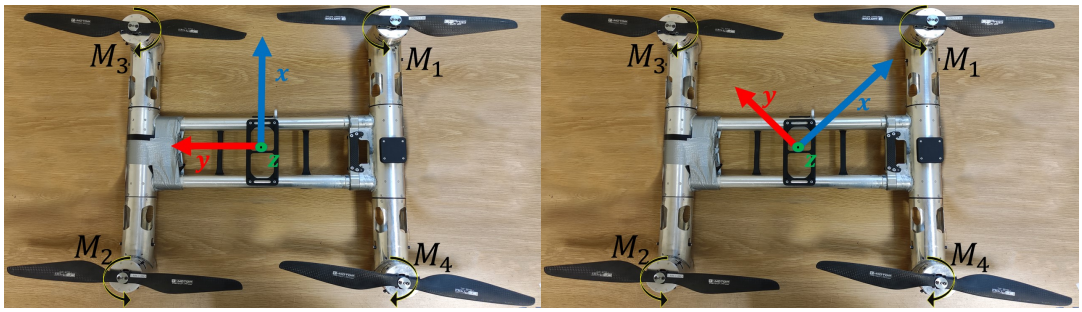
$$Q_i = C_Q \omega_i^2 \tag{A.6}$$

where C_Q is the lumped torque parameter that considers the same conditions true for all the motors.
Let's $u = \left[\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2\right]$ as the vector of the inputs to the 4 actuator modules of the quadrotor with fixed spin direction for each motor group. The resultant control force $f_c \in \mathbb{R}^3$ and control torque $\tau_c \in \mathbb{R}^3$ applied to the centre of mass, are produced according to the following:

$$f_c = B_f u \tag{A.7}$$

$$\tau_c = B_\tau u \tag{A.8}$$

where B_f and B_τ depends on a particular configuration considered. Let us consider the enumeration of the motors groups according to the convention adopted in the PX4 autopilot used in the Alcan02. Considering a configuration in which the x-axis of the body frame points to the first motor group, the y-axis points to the motor group 3, and the z-axis is parallel to the motor axis of rotation, we can define the



(A) (B)

FIGURE A.1: *The images show the frame convention of a quadrotor. In* (A) *the so-called "x" configuration. In* (B) *the so-called "+" configuration.*

+-configuration by the following:

$$B_f := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -C_T & -C_T & -C_T & -C_T \end{bmatrix} \tag{A.9}$$

$$B_\tau := \begin{bmatrix} -C_T r_+ & 0 & C_T r_+ & 0 \\ 0 & C_T r_+ & 0 & -C_T r_+ \\ -C_Q & C_Q & -C_Q & C_Q \end{bmatrix} \tag{A.10}$$

where r_+ is the arm length from quadrotor hub centre to one motor group. Considering a configuration in which the x-axis is perpendicular to the line passing through the motor group 1 and the motor group 3, the y-axis is perpendicular to the line passing through motor group 2 and motor group 3, and the z-axis is parallel to the motor axis of rotations, we can define the *x-configuration* by the following:

$$B_f := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -C_T & -C_T & -C_T & -C_T \end{bmatrix} \tag{A.11}$$

$$B_\tau := \begin{bmatrix} -C_T r_x & C_T r_x & C_T r & -C_T r_x \\ -C_T r_x & -C_T r & C_T r_x & C_T r_x \\ -C_Q & C_Q & -C_Q & C_Q \end{bmatrix} \tag{A.12}$$

where $r_x = r_+ \sin 45$ is the distance between a motor group axis of rotation and the body frame z-axis.

B AlCan02 Additional Protocol Information

The AlCan02 can be thought of as the composition of three modules: the central module, the CANSX, and the CANDX. The CANSX is in charge of taking and processing all the data coming from the other two modules. In particular, the Arduino mega pro 2560, inside the CANSX, is in charge of collecting all the telemetry data from the autopilot, the payload, the motors, ESC and batteries. It communicates the data with a certain frequency and protocol to the MRK1400 via I2C. The MRK1400 collects the data and writes files on the SD card via SPI protocol. When a 3g connection is available, the MRK1400 dumps the files to the server of the drone database via MQTT protocol. When the SD card is almost full, the MRK1400 removes the oldest files to make space for new logs. The Mega2560 sends packets via I2C when available. They consist of a header to communicate the type of data, a variable-length payload and the CRC16.
The possible messages are:

- **Battery**: Header: 0x00 Timestamp (ms UNIX epoch) [uint64] Battery voltage (V) [float32] Current (A) [float32] CRC16 [16 bits]
- **GPS**: Header: 0x01 Timestamp (ms UNIX epoch) [uint64] LAT (degrees *E7 WGS84) [int32] LON (degrees *E7 WGS84) [int32] ALT (meters WGS84) [float32] CRC16 [16 bits]
- **Motor Status**: Header: 0x02 Timestamp (ms UNIX epoch) [uint64] ID motor (N) [uint8] Temperature Battery (degrees C) [uint8] Temperature Motor (degrees C) [uint8] RPM (rpm) [uint16] Current (mA) [uint16] CRC16 [16 bits]
- **Attitude**: Header: 0x03 Timestamp (ms UNIX epoch) [uint64] Roll (degrees) [float32] Pitch (degrees) [float32] Yaw (degrees) [float32] CRC16 [16 bits]
- **Battery UID**: Header: 0x04 Timestamp (ms UNIX epoch) [uint64] ID motor (N) [uint8] UUID (N) [uint32] CRC16 [16 bits]
- **ARTVA payload**: Header: 0x05 Timestamp (ms UNIX epoch) [uint64] ID beacon (N) [uint8] Distance (m) [float32] Angle (degrees) [float32] CRC16 [16 bits]
- **RECCO payload**:Header: 0x06 Timestamp (ms UNIX epoch) [uint64] Power antenna emitter (N) [uint8] Power signal received (N) [uint16] CRC16 [16 bits]

- **Deploy**: Header: 0x07 Timestamp (ms UNIX epoch) [uint64]
- **Autopilot status**: Header: 0x08 Bitmask Faults [16 bits] CRC16 [16 bits]

Once received the data on I2C, the MKR1400 can: write log files on the SD card, open a new log file after the old one exceeds XX Mb or YY seconds passed, delete the older log file is remaining space is < ZZ Mb, saves data in string or binary protocol, depending on a parameter.
The files are logged with the following specification:

- **Header1**: 8 chars for number of bytes written to the database via mqtt (with leading zeros or empty spaces) + "\r \n";
- **Header2**: "UUID drone [uint32] + \r \n";
- **Data (in string protocol)**: each line contains timestamp (ms)[uint32], ID of the data (N, as header described before)[uint16], variable length payload and "\r \n". Example: "52837,1,44.125673,12.1234523,15.5\r\n" for GPS data. Mapping 1:1 with the data received via I2C from the Mega2560;
- **Data (in binary protocol)**: it's the data coming via I2C protocol, without the CRC and terminating with "\r\n".

The data is sent from the drone via 3g network when available. When 2g or 3G connectivity is available, the MRK dumps log files to the DroneDB server via MQTT protocol. The data sent is 1:1 with the data on the SD card. Only full lines are sent, terminating with \r\n. Multiple lines can be sent in a single MQTT packet.
If data on SD is saved with string protocol, also the data sent via MQTT will be string protocol. If data on SD is saved with binary protocol, also the data sent via MQTT will be binary protocol. The MQTT parser should detect the protocol and parse it accordingly. The parsed data is then stored in the DB as time series, ready to be visualized with *Grafana UI*.
A maximum of 500 bytes of data is sent every 500ms.
An API suite running on the DroneDB server allows simplifying data collection by parsing standard flight log (ulog) files retrieved from the flight control board of the drones. All the relevant information is therefore stored in the application database structure using the flight control ID as drone identifier and arming start time as flight identifier. The database structure is as in fig. B.1. All the tables store the appropriate objects in the back-end application. Typically, each drone is associated with one rescue team, that also use a number of batteries. The recorded flight logs are then parsed and associated with each active drone. Each flight log contains the full parameter list, some informative messages and many timeseries data according to the telemetry data stored in the log itself. These timeseries are then visualized using the Grafana reporting tool (or any other graph visualization application). The database can also be populated from the front-end or via dedicated API for administrators. The API and functions available are:

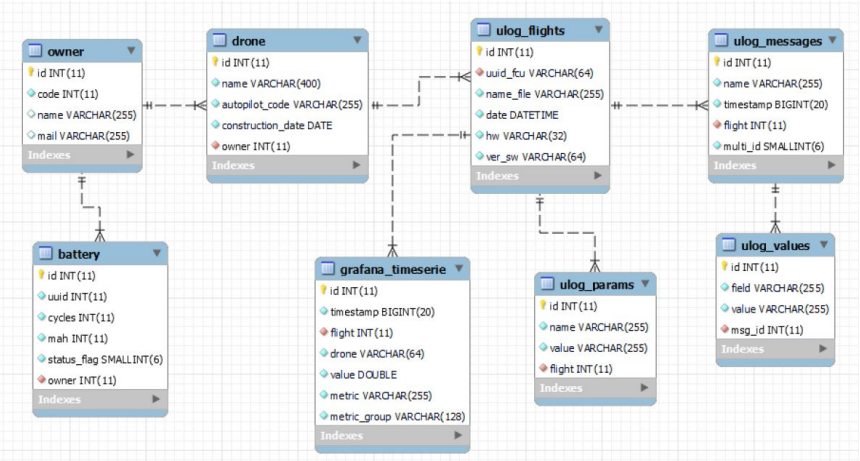


FIGURE B.1: *Drone database's structure.*

- \api\battery\create – create a new battery object
- \api \battery\info – retrieve info on an existing battery object
- \api\battery\list – list available battery objects
- \api\battery\modify – update values of an existing battery object
- \api\battery\delete – delete a battery object
- \api\drone \create – create a new drone platform object
- \api\drone\info – retrieve information on an existing drone object
- \api\drone\list – list available drone objects
- \api\drone\modify – update values of an existing drone object
- \api\drone\delete – delete a drone object
- \api\drone_data\flight – create a new flight object (directly from drone)
- \api\drone_data\data – store a new data log (directly from drone)
- \api\owner\index – owner dashboard
- \api\owner\create – create a new owner team object
- \api\owner\info – retrieve information on an existing owner object
- \api\owner\list – list available owner objects
- \api\owner\modify – update values of an existing owner object
- \api\owner\delete – delete an owner object
- \api\ulog\parse – parse a new flight log (ulog) file

To have a feedback on flight logs, *Grafana* can be used to show graph data of the different log timeseries. It can be configured to create different dashboards that can show all the relevant data about the involved drones of every single team. The user can then apply filters and change time frames to check specific parts of different flights. Each value is stored using its timestamp as the main key to plot the information in the graph tool.

C ARVA Working Principle

The commercially available avalanche transceivers have two operating modes, namely they can work as receivers or as transmitters. A manual switch can be used to commute between the two. The transceivers worn eventual victims are always commuted in transmission mode. The device in receiver mode provides two outputs denote by d and δ which respectively indicates the estimated distance transmitter-receiver and the magnetic vector field orientation projected on the receiver reference frame.
The commercial ARVA devices are composed of 3 different ferromagnetic core geometries that imply different reading efficiency. This makes it easier to build compact and transportable devices but it also imposes the presence of a bigger and primary antenna with a higher signal to noise ratio and smaller secondary and thirdly antennas. This determines the search methods described in section 2.1.1. The bigger antenna can efficiently sense a presence of an EM field far from the transmitter. The remaining two antennas are exploited only when the transmitter is sufficiently close, in order to process the outputs.
In order to describe the magnetic vector field generated by the receiver and sensed by the transmitter, we introduce the following notation. We indicate as $F_f(O_f, x_f, y_f, z_f)$ the reference frame attached to the ARVA transmitter. The axes of the reference frame are aligned with the axes of the three antennas of the device. Additionally, we indicate as $F_r(O_r, x_r, y_r, z_r)$ the reference frame attached to the ARVA receiver, with the frame axis aligned with the three antenna's axis. Finally, with $F_i(O_i, x_i, y_i, z_i)$ we indicate the inertial reference frame. We can indicate with p_t^*, p_r^* respectively the position of the transmitter and the receiver respect to F_* with $* \in i, r, t$. According to [12] we can derive the mathematical model of the magnetic vector field generated by the ARVA in the transmitter mode by

$$h_t^t(p_t) = \frac{||m||}{4\pi||p_r^t||^5} A(p_r^t) \tag{C.1}$$

where

$$A(p_r^t) = \begin{bmatrix} 2x^2 - y^2 - z^2 \\ 3xy \\ 3xz \end{bmatrix}$$

m is the dipole moment according to [44], (x, y, z) are the component of the vector p_r^t. Is possible to express the nominal electromagnetic field in F_i by

$$h_0^i(p_t) = R_t^i h(R_i^t(p_r^i - p_t^i)) \tag{C.2}$$

where R_t^i and R_i^t are respectively rotation matrices from F_t to F_i and vice-versa.
The magnetic field reded from the receiver is indicated as h^r and is given projecting h_0^i onto F_r according to

$$h^r = R_i^r h_0^i + Q^r R_i^r w \tag{C.3}$$

where w^i indicates the noise due to EMI, $Q = diag(q_1, q_2, q_3)$ with $q_i \in (0, 1]$ for $i = 1, 2, 3$ indicate the noise sensitivity of the i-th antenna.
In processing the outputs, the ARVA receiver uses an internal logic that exploits the measured h^r to cluster the search area in three zones, indicative of the distance of the victim. Following the work of Cacace, Mimmo, and Marconi [11] we denote the following. The *outer zone* is characterized by a SNR given by a large distance transmitter-receiver that makes h_r too weak to extract reliable information. Accordingly, the outputs are set to $(d, \delta) = (-1, -1)$. The *mid zone* is characterized by a consistent h_r sensed only by exploiting the bigger and main antenna. The output d is a function of the relative orientation between receiver and transmitter and is given by

$$d = \left(\frac{k_q}{4\pi ||L_{xy} h^r||} \right)^{1/3} \tag{C.4}$$

where L_{xy} is the projector operator on the $x_r - y_r$ axes and k_q is a constant usually tuned by manufacturers. The angle $\delta \in (-90, 90) deg$ is given by

$$\delta = \tan^{-1} \left(\frac{L_y h^r}{L_x h^r} \right) \tag{C.5}$$

where L_x and L_y are respectively projectors operators on the x_r and y_r axis.
Finally, the *inner zone* is determined by a strong h^r that allows to exploits also the secondary and thirdly antennas. As a consequence, the distance d is no more a function of the relative orientation between the transmitter and the receiver and the angle δ is no longer provided. In this scenario, the distance is provided by

$$d = \left(\frac{k_q}{4\pi ||h^r||} \right)^{1/3} \tag{C.6}$$

It is worth noticing that, the determination of the particular zone depends on the real distance receiver-transmitter and the particular device used. During Airborne we ended up using consistently an ARVA Ortovox M2 and occasionally an ARVA Mammut Barryvox delimiting a *mid zone* with $d \in [\underline{d}, \bar{d}]$ where $\underline{d} = 4500cm \pm 500cm$ and $\bar{d} = 40000cm \pm 5000cm$. The *inner zone* is delimited for $d < \underline{d}$ and the *outer zone* for $d > \bar{d}$.
Differently from the commercial devices, the 3D ARVA antenna described in section

2.1.1 works only in receiver mode. Besides, the antenna presents 3 equally sensitive ferromagnetic coils that act as equally efficient antennas in reading the magnetic field intensity. Additionally, the different internal logic produces different outputs than commercial devices. The outputs are the single axis current-related signals properly filtered and amplified. In fact, the internal logic is composed of analogical filters, while the Ortovox M2 and the Mammut Barryvox presents digital filtering. The three outputs, namely (x, y, z) are processed by an onboard algorithm of the Alcan02, as described in Chapter 4.

D PX4 Controllers and Planner

This thesis emphasizes the implementation of high-level guidance algorithms that generate waypoints (in section 5.1) or trajectories (section 5.3.2). The control part is left to the low-level controller of the autopilot PX4. Additionally, until the integration aboard of the algorithm presented in section 5.3.2 the waypoints generated by the algorithm in section 5.1 are connected using the PX4 internal trajectories generator.
The controller is a part of the flight stack, a collection of guidance, navigation and control algorithms for UAVs. The controller takes setpoints, measurements or estimates states as inputs and produces as outputs eventual corrections to reach the setpoints. The outputs are turned into force commands which are translated into individual motor commands by the Mixer. Moreover, it also ensures that some limits are not exceeded. A complete overview of the Flight Stack can be seen in figure D.1. The PX4 controllers are a standard cascaded control architecture where the

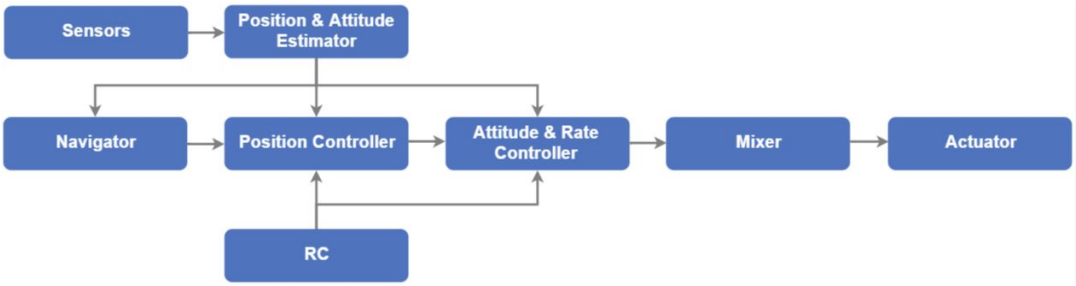


FIGURE D.1: *Overview of the collection of modules that compose the Flight Stack of the PX4.*

controllers are a mix of P and PID. In detail, the overall control architecture that comprehends Position Controller, Attitude and Rate Controller, and Mixer can be seen in figure D.2. The position and velocity controller takes as input the position setpoint r_{sp}, a full 6D pose message of type *geometry_msgs/PoseStamped* using just the data regarding the position. The velocity setpoint v_{sp} is a 6D pose message of type *geometry_msgs/Twist* that only uses the *Vector3 linear* part of the message. The setpoints are used to obtain the desired velocity v_{des} according to:

$$v_{des} = v_{sp} + P(r_{sp} - \hat{r})$$

where $\hat{r}$ is the estimated position. The velocity reference is then passed to the velocity controller and used to obtain the desired force F_sp using the PID according to

$$F_sp = PID(\hat{v}, v_{des})$$

where $\hat{v}$ is the estimated velocity. It is worth pointing out that the PID controllers are performed in the local NED reference frame. The gain in the x and y directions are identical, the gain in the z directions is different. This gives priority to the attitude control by skipping the control in the $x-y$ plane if the control in z saturates i.e. the drone reaches the thrust limit. Moreover, the thrust limits set the anti-windup parameters in the velocity PID. The plane velocity (i.e. the velocity in the $x-y$ plane) control is executed after the z velocity control to give priority to the altitude control since the horizontal thrust is related to the vertical thrust through the tilt angle. All the information concerning the implementation can be consulted on the PX4 firmware location in `src/modules/mc_pos_control`. All information concerning the notation in figure D.2, figure D.3, and figure D.4, can be found at [53].

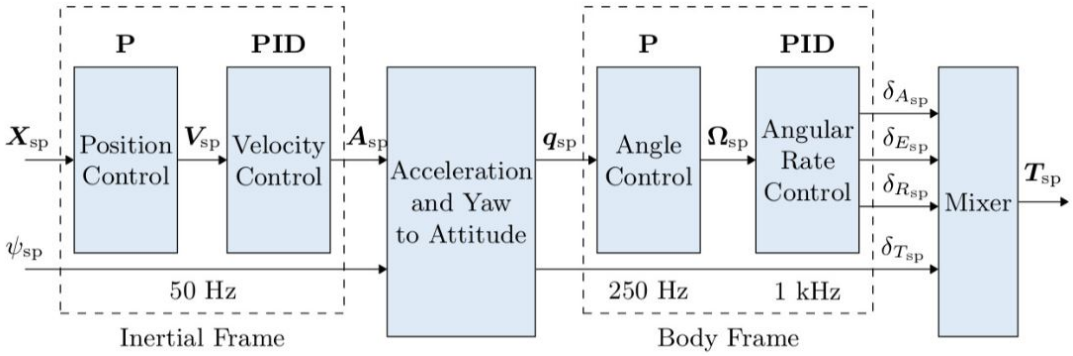


FIGURE D.2: *Multicopter control architecture.*

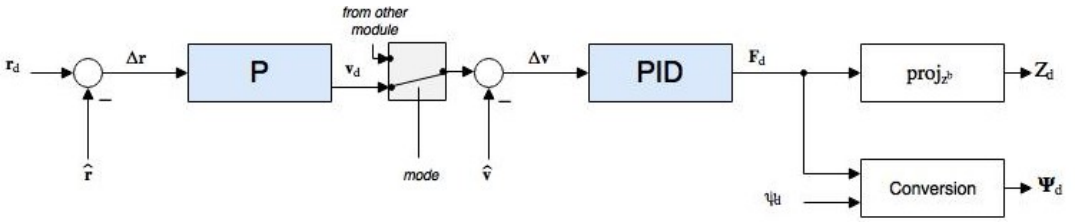


FIGURE D.3: *Multicopter position control architecture.*

The attitude controller is based on the work of Brescianini, Hegn and D'Andrea [8]. Every rotation is parametrized by a single rotation about a fixed axis described by a unit vector $\vec{k}$ and a rotation angle α. Based on this we can define a unit quaternion as

$$\mathbf{q} = [q_0, q_1, q_2, q_3]^T = \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix} = \begin{bmatrix} \cos\frac{a}{2} \\ \vec{k}\sin\frac{a}{2} \end{bmatrix} \tag{D.1}$$

The desired angular velocity $\vec{\Omega}_{sp}$ is computed following the control law:

$$\vec{\Omega}_{sp}(\mathbf{q}) = 2P\,\mathrm{sgn}(q_{0,e})\mathbf{q}_{1:3,e}, \qquad \mathrm{sgn}(q_{0,e}) = \begin{cases} 1, & q_{0,e} \geq 0 \\ -1, & q_{0,e} < 0 \end{cases} \tag{D.2}$$

where $\mathbf{q}$ is the estimated attitude quaternion, $\mathbf{q}_e := \mathbf{q}^{-1} \cdot \mathbf{q}_{sp}$ is the error measure i.e. the rotation from $\mathbf{q}$ to $\mathbf{q}_{sp}$, and P is a tuning parameter.
It is worth noticing that the controller must satisfy

$$\vec{\Omega}_{sp}(\mathbf{q}) = \vec{\Omega}_{sp}(-\mathbf{q}) \tag{D.3}$$

because any physical attitude in $SO(3)$ corresponds to two antipodal quaternions in $\mathbb{S}^3$. It is trivial to prove that, since $\mathbf{q}$ and $-\mathbf{q}$ represent the same attitude value, also the control output should be the same. The value of the output is constrained in order to saturate the rate command. The Attitude Controller can be seen in figure D.4. The Rate Controller is the inner-most loop based on 3 independent PID that control

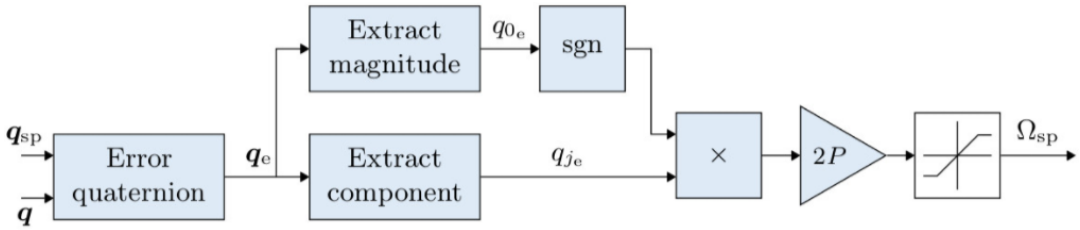


FIGURE D.4: *Multicopter Attitude control architecture.*

the yaw, pitch, and roll body rates. The outputs are fed in the Mixer and are usually limited between -1 and 1. The integral is limited to prevent wind up phenomena. A low-pass filter is used on the derivative path to reduce the noise coming from the rates. The estimation comes from the IMU gyros according to the following pipeline *gyro data -> parameters calibration -> remove estimated bias -> notch filter -> low-pass filter -> vehicle angular velocity -> derivative -> low-pass filter-> vehicle angular acceleration used by the controller*. The controller can be seen in figure D.5.

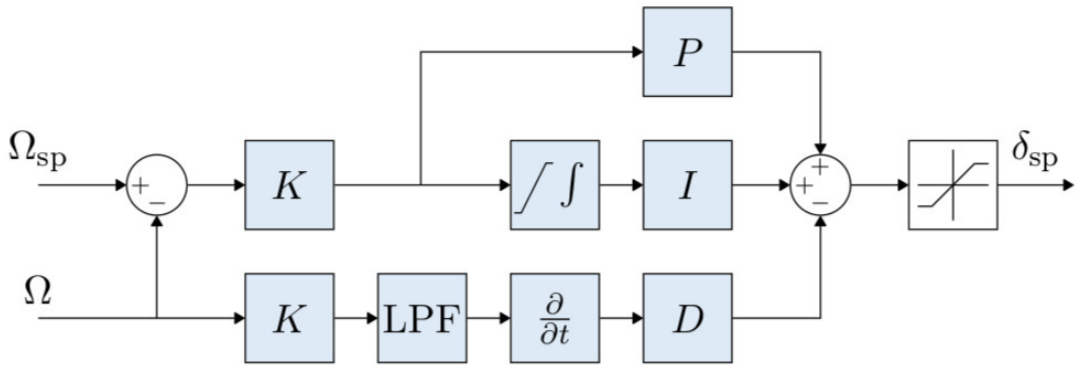


FIGURE D.5: *Multicopter rate control architecture.*

The trajectory generator, implemented in the PX4 firmware, provides jerk-limited

trajectories in response to stick inputs of the RC or mission changes. The planner generates symmetric smooth S-curves that guarantee bounds on jerk and acceleration values. The user can configure the following constraints:

- Maximum jerk j_{Max};
- Maximum acceleration a_{Max};
- Initial acceleration a_0;
- Final acceleration a_3;
- Initial velocity v_0;
- Desired velocity v_0.

An example of a jerk, acceleration, velocity, and position profiles can be seen in figure D.6.

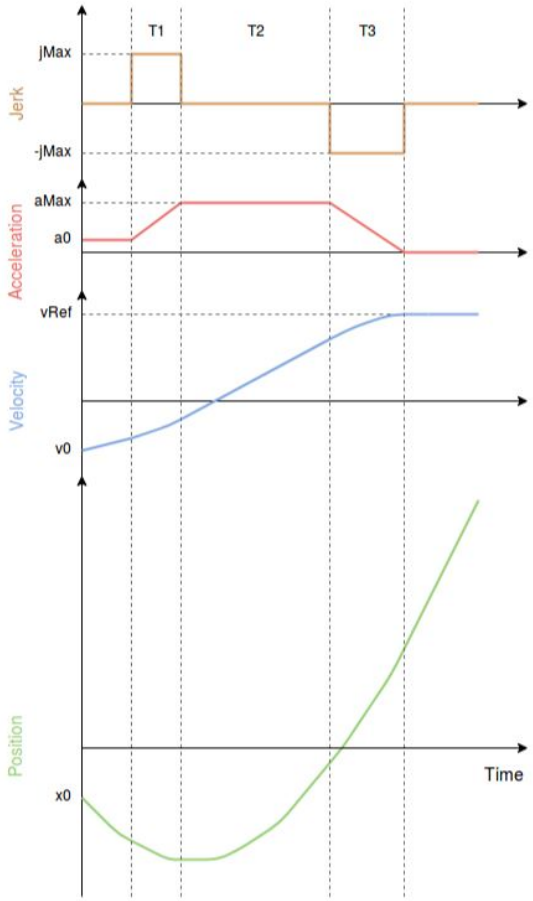


FIGURE D.6: *Example of profiles generated by the PX4 jerk-limited planner*

Bibliography

[1] Ali Ameri et al. "Regression convolutional neural network for improved simultaneous EMG control". *Journal of Neural Engineering* 16 Mar. 2019. DOI: 10.1088/1741-2552/ab0e2e.

[2] Ilario Antonio Azzollini et al. "UAV-Based Search and Rescue in Avalanches using ARVA: An Extremum Seeking Approach". *arXiv preprint arXiv:2106.14514* 2021.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". *ArXiv* 1409 Sept. 2014.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 2013, pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[5] Luigi Biagiotti and Claudio Melchiorri. *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[6] Patrick Bouffard and Steven Waslander. "A Hybrid Randomized/Nonlinear Programming Technique For Small Aerial Vehicle Trajectory Planning in 3D" Jan. 2009.

[7] Dario Brescianini and Raffaello D'Andrea. "Computationally efficient trajectory generation for fully actuated multirotor vehicles". *IEEE Transactions on Robotics* 34.3 2018, pp. 555–571.

[8] Dario Brescianini, Markus Hehn, and Raffaello D'Andrea. *Nonlinear quadrocopter attitude control: Technical report*. Tech. rep. ETH Zurich, 2013.

[9] Lorenzo Marconi Bruno Strano. "Regressive Convolutional Neural Network for ARTVA detection" 2021.

[10] J. Cacace et al. "A control architecture for multiple drones operated via multimodal interaction in search amp; rescue mission". *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2016, pp. 233–239. DOI: 10.1109/SSRR.2016.7784304.

[11] Jonathan Cacace, Nicola Mimmo, and Lorenzo Marconi. "A ROS Gazebo plugin to simulate ARVA sensors". *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020, pp. 7233–7239. DOI: 10.1109/ICRA40945.2020.9196914.

[12] Jonathan Cacace, Nicola Mimmo, and Lorenzo Marconi. "An ARVA sensor simulator". *Robot Operating System (ROS)*. Springer, 2021, pp. 233–266.

[13] John Canny and John Reif. "New Lower Bound Techniques for Robot Motion Planning Problems". *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*. SFCS '87. USA: IEEE Computer Society, 1987, pp. 49–60. ISBN: 0818608072. DOI: 10.1109/SFCS.1987.42. URL: https://doi.org/10.1109/SFCS.1987.42.

[14] Tao Cheng, Chunhui Liu, and Wenrui Ding. "Weak Signal Detection Based on Deep Learning". May 2019, pp. 114–118. ISBN: 978-1-4503-7171-1. DOI: 10.1145/3330393.3330409.

[15] Ronald Choe et al. "Cooperative trajectory generation using pythagorean hodograph Bézier curves". *Journal of Guidance, Control, and Dynamics* 39.8 2016, pp. 1744–1763.

[16] Venanzio Cichella et al. "Optimal Multivehicle Motion Planning Using Bernstein Approximants". *IEEE Transactions on Automatic Control* 66.4 2020, pp. 1453–1467.

[17] Thomas F Coleman and Yuying Li. "A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables". *SIAM Journal on Optimization* 6.4 1996, pp. 1040–1058.

[18] Riccardo Costanzi et al. "Generic path planning algorithm for mobile robots based on Bézier curves". *IFAC-PapersOnLine* 49.15 2016, pp. 145–150.

[19] Hoang Anh Dau et al. *The UCR Time Series Archive*. 2019. arXiv: 1810.07758 [cs.LG].

[20] Hoang Anh Dau et al. "The UCR time series archive". *IEEE/CAA Journal of Automatica Sinica* 6.6 2019, pp. 1293–1305.

[21] Francesco De Giudici et al. "Design and testing of an autonomous ARTVA detector for small drones". *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. IEEE. 2021, pp. 104–108.

[22] Lee Fowler and Jonathan Rogers. "Bézier curve path planning for parafoil terminal guidance". *Journal of Aerospace Information Systems* 11.5 2014, pp. 300–315.

[23] Franco Fuschini et al. "An UAV-Based Experimental Setup for Propagation Characterization in Urban Environment". *IEEE Transactions on Instrumentation and Measurement* 70 2021, pp. 1–11. DOI: 10.1109/TIM.2021.3104401.

[24] John Cristian Borges Gamboa. *Deep Learning for Time-Series Analysis*. 2017. arXiv: 1701.01887 [cs.LG].

[25] Fei Gao et al. "Online Safe Trajectory Generation for Quadrotors Using Fast Marching Method and Bernstein Basis Polynomial". *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 344–351. DOI: 10.1109/ICRA.2018.8462878.

[26] Philip E Gill, Walter Murray, and Margaret H Wright. *Practical optimization*. SIAM, 2019.

[27] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review". *Data Mining and Knowledge Discovery* 33.4 2019, pp. 917–963. ISSN: 1573-756X. DOI: 10.1007/s10618-019-00619-1. URL: http://dx.doi.org/10.1007/s10618-019-00619-1.

[28] J.F.Canny. "The Complexity of Robot Motion Planning." *MIT Press* 1988.

[29] Sertac Karaman and Emilio Frazzoli. *Sampling-based Algorithms for Optimal Motion Planning*. 2011. arXiv: 1105.1186 [cs.RO].

[30] Emre Koyuncu and Gokhan Inalhan. "A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3D environments". *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008, pp. 815–821. DOI: 10.1109/IROS.2008.4651122.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". *Neural Information Processing Systems* 25 Jan. 2012. DOI: 10.1145/3065386.

[32] Steven M LaValle et al. "Rapidly-exploring random trees: A new tool for path planning" 1998.

[33] Yann LeCun, Y. Bengio, and Geoffrey Hinton. "Deep Learning". *Nature* 521 May 2015, pp. 436–44. DOI: 10.1038/nature14539.

[34] Pankaj Malhotra et al. "Long Short Term Memory Networks for Anomaly Detection in Time Series". *ESANN*. 2015.

[35] L. Marconi et al. "The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments". *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. 2012, pp. 1–4. DOI: 10.1109/SSRR.2012.6523905.

[36] Phillipe Martin, Richard M Murray, and Pierre Rouchon. "Flat systems, equivalence and trajectory generation" 2003.

[37] Syed B Mehdi et al. "Collision avoidance through path replanning using Bézier curves". *AIAA Guidance, navigation, and control conference*. 2015, p. 0598.

[38] Daniel Mellinger and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors". *2011 IEEE international conference on robotics and automation*. IEEE. 2011, pp. 2520–2525.

[39] Nicola Mimmo, Pauline Bernard, and Lorenzo Marconi. "Avalanche victim search via robust observers". *IEEE Transactions on Control Systems Technology* 29.4 2020, pp. 1450–1461.

[40] Julien Moreau et al. "Reactive Path Planning for Autonomous Vehicle Using Bézier Curve Optimization". *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 1048–1053. DOI: 10.1109/IVS.2019.8813904.

[41] Julien Moreau et al. "Reactive path planning for autonomous vehicle using Bézier curve optimization". *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2019, pp. 1048–1053.

[42] Roberto Naldi, Michele Furci, and Lorenzo Marconi. "Modeling and control of a class of multi-propeller aerial vehicles". *IFAC Proceedings Volumes* 46.30 2013, pp. 7–12.

[43] Michiel Van Nieuwstadt and Richard M. Murray. "Real Time Trajectory Generation for Differentially Flat Systems". *IFAC Proceedings Volumes* 29 1996, pp. 2301–2306.

[44] P. Pinies and Juan Tardos. "Tardos, "Fast localization of avalanche victims using sum of Gaussians". June 2006, pp. 3989–3994. DOI: 10.1109/ROBOT.2006.1642314.

[45] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Vol. 3. Cambridge university press, 1999.

[46] Mario Silvagni et al. "Multipurpose UAV for search and rescue operations in mountain avalanche events". *Geomatics, Natural Hazards and Risk* 8.1 2017, pp. 18–33.

[47] Ezra Tal and Sertac Karaman. "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness". *IEEE Transactions on Control Systems Technology* 29.3 2020, pp. 1203–1218.

[48] Michiel J Van Nieuwstadt and Richard M Murray. "Real-time trajectory generation for differentially flat systems". *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 8.11 1998, pp. 995–1020.

[49] Qingsong Wen et al. "Time Series Data Augmentation for Deep Learning: A Survey". *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence* 2021. DOI: 10.24963/ijcai.2021/631. URL: http://dx.doi.org/10.24963/ijcai.2021/631.

[50] Yinyu Ye and Edison Tse. "An extension of Karmarkar's projective algorithm for convex quadratic programming". *Mathematical programming* 44.1 1989, pp. 157–179.

Sitography

[51] *Airborne project website*. URL: https://www.airborne-project.eu/.

[52] *ARVA RCNN dataset*. URL: https://github.com/StranoB/RCNN-ARTVA-Detection/.

[53] *PX4 firmware website*. URL: https://docs.px4.io/master/en/.

[54] *RECCO website*. URL: https://recco.com/.

[55] *Sherpa project website*. URL: http://www.sherpa-fp7-project.eu/sherpa/.

[56] *Xlog website*. URL: http://x-log.de/.