Heterogeneous Swarm
Hardware Integration and Deployment

Tristan LE FLOCH
ROB 2024

Acknowledgements

I would like to acknowledge and give my warmest thanks to my supervisor Dr Philip Anderson who made this work possible. His guidance and advice carried me through all the stages of my internship, thanks for your brilliant comments and suggestions. I would also like to thank the HR team who did a wonderful work and helped me a lot through the administrative process of the visit, and I also thank all SAMS members for welcoming me, and for letting my stay be an enjoyable moment.

Abstract

My internship went through 3 main stages. First I discovered SAMS and the equipment I would be working with: autopilots, Impyacks, drones, ... I had to familiarise with the platforms and learn the prerequisites to achieve the objectives of the project. Then, I worked a lot with the already existing equipment to improve it and integrate new pieces on it. Finally, I focused on the building of a rover and the electronics inside, which was planned to be used for educational and training purposes. The aim was to make this rover autonomous with a Pixhawk.

Contents

1 Introduction

1.1 Scottish Association for Marine Science (SAMS)

The Scottish Association for Marine Science (SAMS) is a non-lucrative marine science organisation acting as a research institute. As a partner of the University of Highlands and Islands (UHI), SAMS also provides high-level education in Marine Science through various programmes including Marine Science BSc (with Arctic Studies or Oceanography with Robotics streams) and PhD.
SAMS is ideally located near Oban on the west coast of Scotland, the gateway to the inner Hebrides. Its wonderful location and easy access to coastal and offshore environments allow SAMS' members to easily run their researches on the field.

Among the various research facilities at SAMS, one in particular fits into the context of this internship: the Scottish Marine Robotics Facility. The experienced and skilled staff and students at SAMS' Scottish Marine Robotics Facility develop, adapt, deploy and operate a wide range of diving, flying and sailing robotics platforms to answer pressing environmental science questions.

1.2 Context and project

The overall project the Dr Philip Anderson is on is understanding the atmosphere - ocean interface, and heat, momentum and how trace gases are transported through the respective boundary layers, and thence modify those layers. Existing methods are inadequate because they always involve a boat large enough to transport the sensing instruments and their operators, which unavoidably disturb the flows involved in the air-sea interactions. For this precise reason; small, instrumented robotic platforms, operating as a heterogeneous swarm, are a key to gaining insight into these processes..

My role is to use my Mobile Robotics background to operate the robotic platforms as desired.

1.3 Objectives

The challenge in this project is to manage the operation of running three different platforms in a coherent manner, that is a small semi-commercial Autonomous Underwater Vehicle (AUV) named EcoSub and two Ardupilot-based platforms: a 2.5m Autonomous Surface Vehicle (ASV) and a water-landable Remotely Piloted Aircraft System (RPAS).
My task is the instrumentation integration which includes: setting up the communications (MAVLink channels), programming the controllers, testing and learning how best to get the RPAS to 'follow' the ASV in echelon, and with a methodology that is expandable (i.e. more platforms, but not more pilots and flight crew).
Due to the length of the internship, the operation of the AUV is not to be considered.

2 Familiarization with the platforms

2.1 Pixhawk

When we think of an easy-access and easy-integration controller to make autonomous robots follow a desired trajectory or stabilise at a desired position, what comes in mind is the autopilot. Ones of the best, wide-spread autopilots is the Pixhawk series, working with MAVLink packets. They are made very accessible for a user even with few experience in robot control. Initially these autopilots were made for copters but now they can be used (with the adequate firmware) on planes, rovers, boats and even subs. Coupled with a mission planning software such as Mission Planner or QGroundControl, they allow to make autonomous whatever is equipped with a Pixhawk, a battery, a GPS and a few motors (even a simple Tupperware).
Indeed, the Pixhawk devices do perfectly respond to the objective of an easy-access structure which is also easily reproducible as we extend the fleet of robotic platforms. In addition, this choice was made to be coherent with the educational use of the Pixhawk and Mission Planner in SAMS' robotics course.

(a) Pixhawk

(b) Pixhawk 4

Figure 2: Pixhawk autopilots used during the internship

2.2 MissionPlanner

A mission planning software is essential to operate a Pixhawk-based robot. Mission Planner and QGroundControl are the most famous and used mission planning software. Both are relevant to operate the robots used here, but in order to have a good insight on the parameters involved in the control of the robots, Mission Planner is the best. This is why we chose to use Mission Planner as our reference for mission planning and operation interface.

Figure 3: Mission Planner interface

The firmwares available on MissionPlanner are offered by Ardupilot, an open-source series of firmwares designed for autopilots. Initially developed for copters, now it supports many vehicle types such as: multi-copters, traditional helicopters, fixed wing aircraft, boats, submarines, rovers and more.
We use Mission Planner to upload the adequate firmware on the Pixhawk, then we can change its parameters accordingly with the nature of the robot (for example skid-steering or not), and finally for PID tuning.

Figure 4: Ardupilot: Versatile, Trusted, Open

2.3 ImpYacks

One of the two main platforms I worked on during the internship is the "Impyack". The Impyacks are SAMS' autonomous kayaks used for various scientific purposes, especially in marine science because they allow the researchers and PhD students to use specific sensors.

Figure 5: Impyacks in the lab

The Impyacks are equipped with a Pixhawk autopilot, a GPS, a Sik radio for telemetry communication, two brushed motors T200 from BlueRobotics, two ESCs and a battery. These boats work in Skid-steering, which means they will use a differential thrust on both motors to turn instead of having a rudder (or a steering rod on cars) to control the direction. Fortunately, this kind of boat is very similar to rovers for the control algorithm, this is why we uploaded the ArduRover firmware on the Pixhawk. Then, there is just to set the parameter FRAME_CLASS equal to 2 to make the system consider itself as a boat.

2.4 Drone training

One of the main objective of this project, is to operate some water-landable drones flying in echelon (one above the other) and following an Impyack. The ability to "land" on water is relevant in this operation because it allows to insure the security of the drone and to facilitate its deployment.

Figure 6: Water-landable drone

In order to safely and legally pilot these aircrafts for a project carried out at SAMS, a professional drone pilot training was necessary. So I received legal and safety lectures for the compliance with the CAA (Civil Aviation Authority), before passing the CAA's official theory test to get a flyer ID. A subsequent practical assessment on safe operations followed on a suitable clear-weather day, with a DJI Phantom 4 Pro.

3 Hardware integration

3.1 SiK Telemetry Radios

As a swarm, the robots have to communicate between each other, but we also want them to communicate with the ground control station (our laptop) in order to monitor their attitude, position, states of battery and sensors, etc. To communicate between our laptop and the robots, we use a SiK Telemetry Radio; a small, light and inexpensive radio platform which we just have to plug on the Pixhawk and on the laptop, providing the easiest ways to setup a telemetry connection between an autopilot and a ground station. Moreover, its firmware is open-source and is easy to update on Mission Planner for instance. The ones I used work on 433 MHz but there are also other models on 915 MHz for example.

Figure 7: SiK Telemetry Radio

Then, the point with these radios in the project is that we want to deploy several platforms at the same time, and therefore establish several communication links at the same time between each platform and its ground station. In order to respond this issue, we want to modify the parameters of the firmware. Indeed, we can define a channel number within the SiK radio's parameters, which helps us to establish distinct links between the different radios operating at the same time. What we figured out later is that it seems there is interference between the radio connections. In fact, when the telemetry traffic is too important, the average transmit time of

information gets quite high which is not handy for good monitoring and even worse if the robots have to talk to each other. The time to establish the connection between two radios is particularly high when other radios are operating.
Here comes the DUTY_CYCLE parameter. The DUTY_CYCLE is the maximum percentage of time that the radio will transmit packets. By default, it is set to 100 so the radio can communicate 100% of the time. Our hypothesis is that we can reduce it so the telemetry traffic is less busy and the radios can communicate without interfering. So I did some tests.

For this experience I did two series of measurements:

- First, I measure the time to connect from the laptop to the autopilot by radio link for different values of DUTY_CYCLE, without interference
- Then, I do the same with an other radio link operating in the same room with the same DUTY_CYCLE

Here is the result:

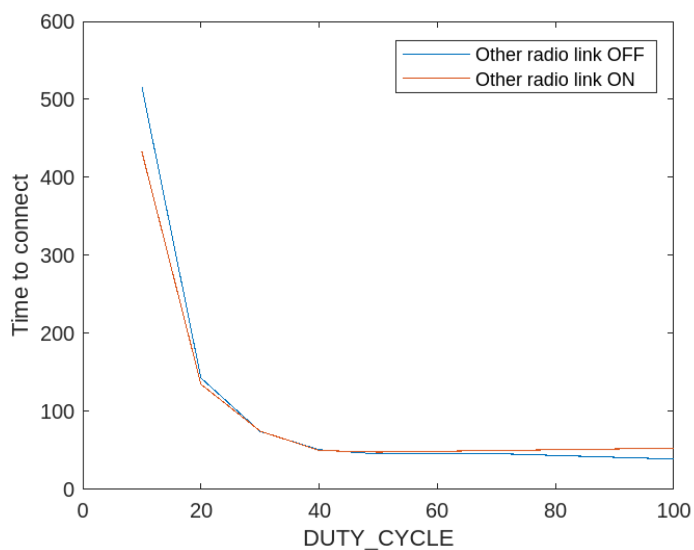


Figure 8: Time to connect with different values of DUTY_CYCLE

We can observe at DUTY_CYCLE 100 that the connection is worse when an other radio link is operating. The difference of time between both curves decreases as the DUTY_CYCLE decreases to reach zero for a DUTY_CYCLE equal to 40. In parallel, the time increases exponentially as the DUTY_CYCLE decreases, what means that a DUTY_CYCLE lower than 40 is not relevant. Moreover, the difference is even more relevant for low values of time, that is to say high values of DUTY_CYCLE.
To conclude, we have to choose the value that is the best compromise between the difference of time and the absolute value of time. For this reason, the best DUTY_CYCLE for our swarm of robots appears to be 40%.

In addition, we obviously need the radio on the robotic platform to send a lot of data to the ground station On the other hand, during an autonomous mission, the ground station is just used to monitor the system but we don't want to send any data except for the start and stop command. Then, we can optimise once again the communication and reduce the interference by reducing the DUTY_CYCLE on the ground station's radio, because it receives a lot of information but doesn't transmit as many packets as the embedded controller.

3.2 Kayaks' propellers

The Impyacks were thought to be easy to take in hand. As they share the same model, they must be equally built, at least on the same base because various sensors were added later for specific uses. In particular, the Impyacks are equipped with two T200 motors from Blue Robotics, we had to check how to best integrate them on the boat. Indeed, two different orientation are possible for the propellers, such a manner that the motor have either to turn clockwise or counterclockwise to go forward in function of the orientation of the propeller. Obviously, if both were turning clockwise (resp. counterclockwise) the cap of the boat would deviate to port (resp. starboard), so they must be oriented differently, one turning clockwise, the other turning counterclockwise.

The question now is: should the right motor turn clockwise or the left one ? From the previous reasoning we know that it is better that a motor is turning clockwise for the boat to steer port. In addition, if we want to steer port, it is easier if the starboard motor is spinning faster because the torque is bigger. For these two reasons combined we choose the starboard motor as the one spinning clockwise when the boat is going forward. For the same reasons, we choose the port motor as the one spinning counterclockwise when the boat is going forward. Therefore, the motors are more efficient with this configuration when the boat goes forward. We had the opportunity to confirm this on the field.

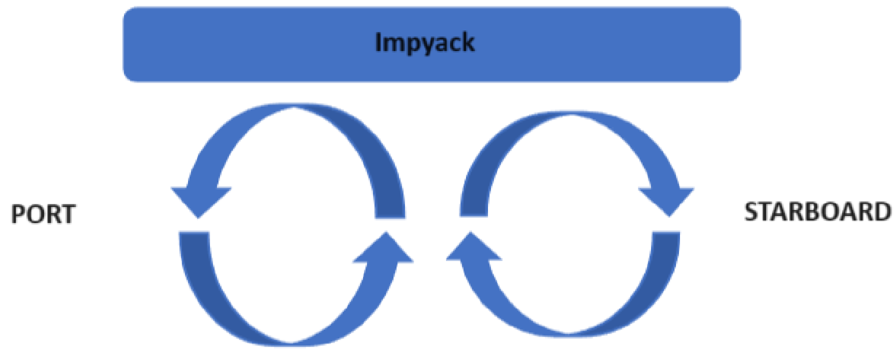


Figure 9: Sens of rotation of the motors when going forward. View from stern

3.3 CO_2 sensors

The main goal of the flying drones, is to measure various physical values such as temperature, relative humidity and quantity of CO_2. The sensors used for this purpose are all integrated in one, and we control it with an Adafruit Feather M0 Adalogger board which also stocks the collected data on a SD-card.
Before engaging a full swarm mission, we want to be sure the values measured by the sensors are correct. We figured out there was an offset in the values of Relative Humidity (RH) and temperature between the two set of sensors. So after a few tests I identified this constant offset in order to correct the values for a coherent analysis after the measurements.

But what interested us the most in these physical values was the quantity of CO_2. The full scale of values of this sensor is 0-10000 ppm. I did a simple test with two 20 minutes sessions of measurements, one without any interaction with the sensors and a second one where I blew on the sensor during 1 minute. We can observe on the Figure 11 that the quantity of CO_2 is pretty low and constant with no perturbation,

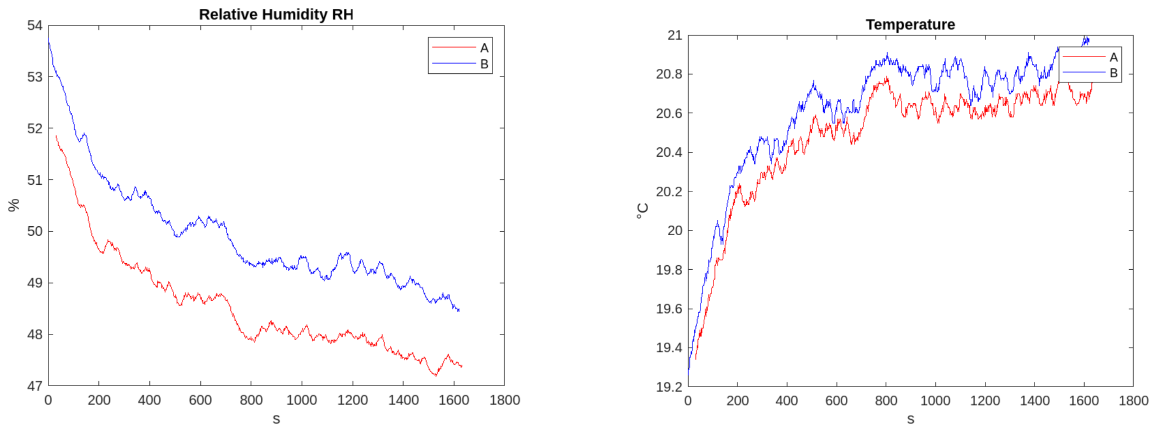
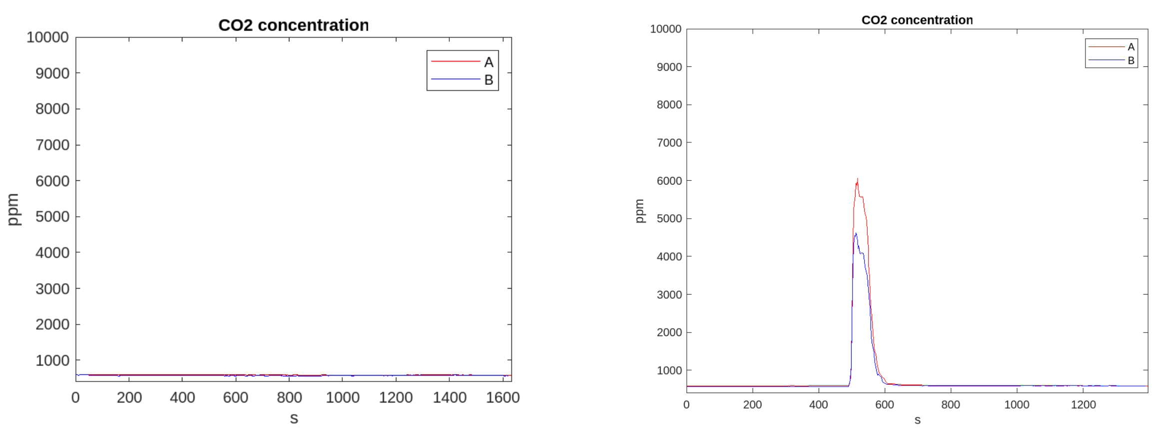


Figure 10: Offset in the measurements of Relative Humidity and Temperature

and we can clearly see a peak when I blow on it. The difference during the blow phase is simply due to the fact that I can't be precise enough to blow equally on both sensors so we don't need to take it into account.

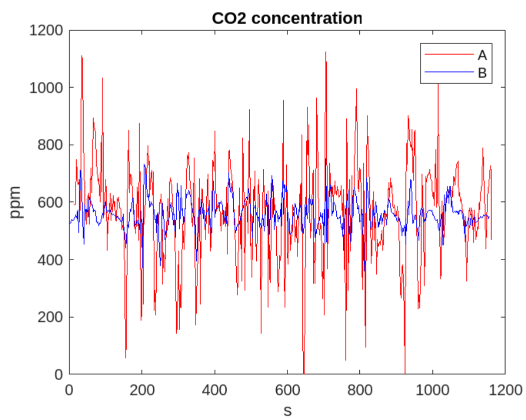


(a) Sensors uncovered

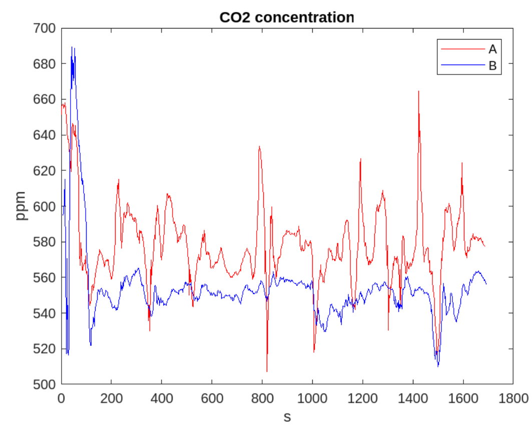
(b) Sensors covered

Figure 11: Measurements of CO_2 inside

In fact, we want to measure quantities of CO_2 outside in natural conditions, that is to say without our direct perturbations. That means that we want to be able to detect and measure very low variations. Let's zoom on a set of data collected outside with no direct interaction.



(a) Sensors uncovered



(b) Sensors covered

Figure 12: Measurements of CO_2 outside

Let's focus first on Figure 12(a), we can see that at this scale, the variations are messy, especially for the sensor A, the values are changing around and average value. We figured out that when we cover the sensor (Figure 12(b)), these variations are much lower but we keep the same average value, that is to say the measurements are still correct, but more relevant: they are better.

4 Deployment

4.1 Settings

The first thing we want to be able to do before launching an autonomous mission with an Impyack, is to control it with a gamepad controller so we can check its good behaviour. We set up a guide to setting up the Impyack and especially its autopilot. This guide goes through 5 steps which are not to be missed for the set up to be accurate:

1. Gamepad paddle allocations:

   We use a simple Logitech gamepad and we need to configure Mission Planner with the right allocations on each ground station.

2. Motor Test:

   This step aims to check the good integration of the propellers and motors, two wires can be inverted in the ESC connections in order to correct the sense of rotation if necessary.

3. Servo Output allocations:

   Simple check of the good allocation and behaviour of the servo output. Nothing should have to be changed.

4. Servo Output Tests:

   First, here we check the whole combination of the previous steps directly with the behaviour of the motors when going forward and turning right/left. Then we check the behaviour when going backward, the same rules should apply except that the motors are turning in reverse. If the behaviour is not the one expected, the user should check that the parameter PILOT_STEER_TYPE is set to 3, this ensures that the system is in skid-steering mode.

5. Other Ardurover parameters to check:

   Finally, if the skid-steer still doesn't work, check the values of the two parameters SERVO1_FUNCTION and SERVO3_FUNCTION which must correspond to the PWM outputs chosen during the hardware integration phase on the Pixhawk.

4.2 Check-list

With the same idea of making the fleet of Impyacks accessible for the researchers and PhD students, we set up a must-go-through Check-list in order to insure the good and safe deployment on the field. As the deployment occurs on a water surface (the sea, a loch, a river), it is hard to reach the robot if a problem occurs and there is not the necessary tools. In order to avoid any issue, this homemade check-list goes through the packing stage, then the on-site set up stage before covering the power-up stage, and finally pre-launch final checks on water.

5 Autopilot on a rover

5.1 Rover for training and education

The deployment of the Impyack requires particularly good conditions: high tides to avoid the seaweed in the propellers, not a rainy weather to not damage the equipment (what happened to be pretty rare in Scotland during Summer), a satisfying satellite count to have a good positioning with the GPS, a manned boat available in the case of an issue occurs on the Impyack, two people available to insure the security of the equipment (one on the boat, one on the beach monitoring and managing the operation),... Indeed, it would be obviously a lot easier to operate on land as some of the previous issues would disappear. Moreover, the teaching in the use of robotic platforms is still one of the main objective of the Impyack project but it would help a lot if the deployment and accessibility was easier. Here came the idea of using a rover with exactly the same control electronics than on the Impyack, so the user can learn how to use an autopilot and a mission planning software, train on the rover before using an Impyack, or simply do some tests on the rover instead of the Impyack when the conditions are not good for it.

Figure 13: Leo Rovers

For this purpose, my supervisor provided me with two four-wheeled rovers from the LeoRover company. I started from zero, so I had to fully mount the robots first.

The electronics is provided for the rovers, there are a Raspberry Pi 4, a camera, a wi-fi emitter and a Leo board. The robot works with ROS for the control and the camera node is also integrated. We can connect to the Wi-fi network emitted by the robot, then view its camera vision and control it with a keyboard or a virtual joystick. Of course, this robot with this provided electronics can be used for a lot of applications as line-following, image recognition,... But our aim is to use it with the Pixhawk with no need of a companion computer, so we removed the provided electronics, and my task now is to provide a new electronic system in order to integrate the Pixhawk and operate the rover the same way we do with the Impyack.

5.2 Skid-steering mode with 4 DC motors

Lero Rover's motors are brushed DC motors which is different from the brushless motors used on the Impyacks. In order to control them with the autopilot, we need speed controllers to understand the PWM commands sent by the controller and then send the corresponding desired voltage to the motors using the power supply, a 12V Li-ion battery. Here comes the Adafruit Feather M0 Adalogger, an Arduino-like controller; and the Adafruit Feather Wing, a shield for the Feather board which allows to control 4 DC motors. It does exactly what we want to, after uploading on the board a program to read the PWM values sent by the autopilot and command the motors with the Library associated to this shield.

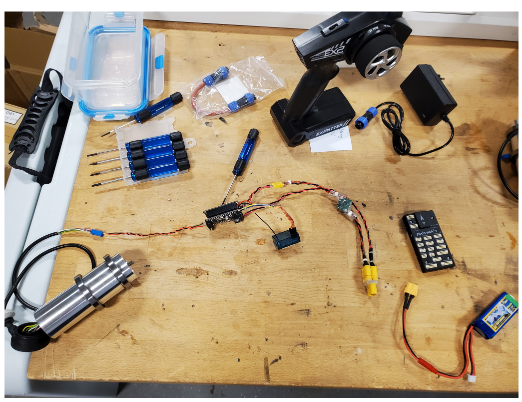


Figure 14: Learning how to use a DC motor with a Feather Wing on a Feather board receiving a PWM command

The Feather controller receives only 2 PWM commands because the rover needs to operate in Skid-steering mode: the right motors receive one same command and the left motors the other one, on each side the 2 motors turn at the same rate and in the same direction. In order to manage and change the direction of the brushed motors, the Feather Wing uses H-bridges.

Once all of this was integrated and set up, I could try it on the field and start the PID tuning before launching a waypoint following mission. Unfortunately, one big issue occurred after using the rover for about 30 minutes. An obstacle slew down one motor which then asked for too much current. What happened is that the back EMF decreased and more current flowed through the H-bridge which burnt out at this moment.

Figure 15: Rover test

To solve the issue, I had to find a better H-bridge that could support more current, so I tried to use the L298N board, a cheap easy-go DC motor driver with a good H-bridge. One L298N board can manage two DC motors so I used one board for each side of the robot.

The point I had trouble with then was the power supply of the L298N board and the Feather M0 Adalogger board. I received the Feather board from my supervisor with its power pins already soldered. I mounted the whole electronic system, then I tried it but nothing happened, the motors didn't work for an obscure reason. So I used a breadboard to do unitary tests on every components and I measured the voltage on the pins to check the commands. Everything worked well individually so I had trouble finding the source of the issue. I finally figured out that the Feather board was not supplied with the necessary power anymore when 4 motors where plugged. The USB pin of the board was supplied with 5V and then regulated to 3.3V as the board needs. In fact, this method was not recommended in the documentation because a retro charging could occur when plugging the laptop to the board and damaging it for example. The two primary ways for powering a Feather board are a 3.7/4.2V LiPo battery plugged into the JST port or a USB power cable plugged into the USB port. So I gave up the USB pin and the main power source of the robot, and tried with the LiPo battery in the JST port. Suddenly, everything worked perfectly well. I could then go on the field.

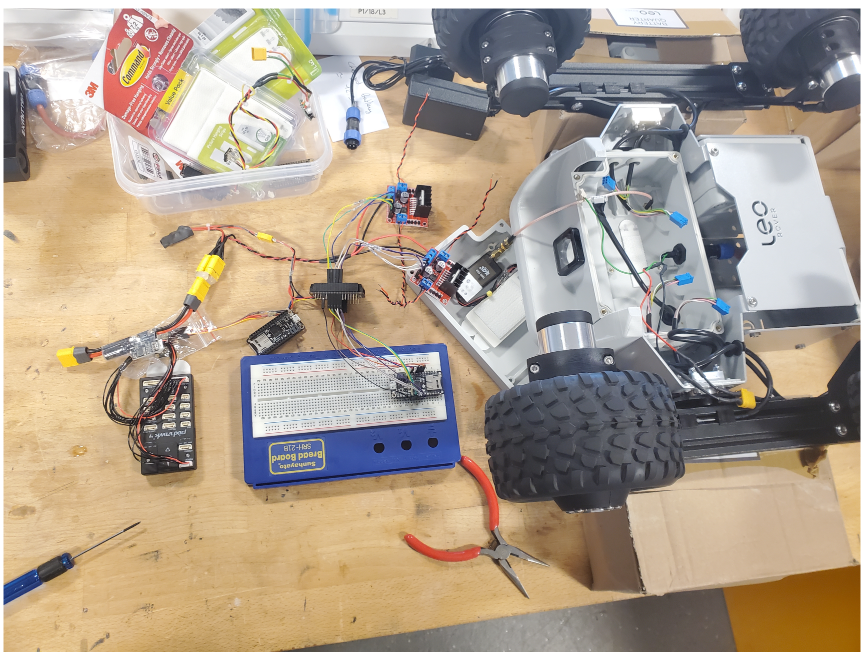


Figure 16: Tests with the L298N boards on a breadboard

The left inconvenient is the fact that the robot now has two power supply and that the little Li-Po battery is not accessible from the outside of the robot as the other one. The solution I recommend but didn't have the time to integrate is to use the 5V regulator and plug the power supply on the 5V and GND pins of a USB Micro B male to 5 pins adapter.

5.3 PID Tuning

I could finally use the rover outside and goes through the PID tuning process on Mission Planner which is very easy to learn on the ArduPilot detailed website.

Figure 17: PID Tuning

When I had been through all the steps and found satisfying coefficients for the PID. I could use Mission Planner to plan a waypoint-following mission, which did work out. I achieved my internship on this success.

Figure 18: Waypoint follow

6 Conclusion

Unfortunately, I didn't reach all the initial objectives of this internship. But actually, they evolved during my stay because we faced various challenges such as the bad weather or difficulties with the providing of equipment. Before this internship I know a lot about the theory in robotics but I had a lack of skills in hardware and electronics. I am glad to have learnt a lot about the electronics and the autopilots through various challenges which I had to face with a lot of autonomy.

Through my visit at SAMS, I also had the opportunity to participate in various meetings with people from other institutes across the United Kingdom. We exchanged about our project and learned from the other. It was a chance to explain the project I was working on and it could potentially interest other people to work with the results.

I am also glad that I have discovered a lot about the local culture and language. I practised a lot my English skills and faced new words and accents that I learned from. I took advantage of the wonderful location to discover the dramatic outer spaces in the Highlands.