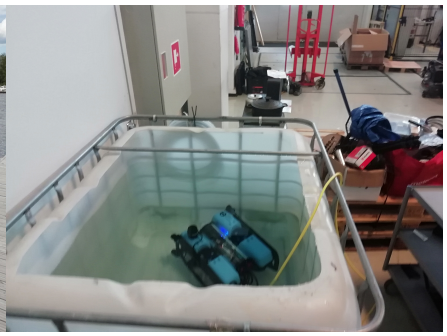
ROS package for BlueROV2

Nathan FOURNIOL

June-August 2019

University of Tampere, Finland

ENSTA bretagne, France



Abstract

This report of my second year internship at Tampere University of Technology deals with the build of a ROS package for the BlueRov2, a remotely operated underwater vehicle. Thus, this report starts with objectives of the aCOLOR project and issues of the role of the BlueRov2 before moving to the creation of a ROS architecture and codes in Python to began the automation of the robot. Cyril and I worked on depth and heading controller before facing difficulties to estimate velocity with two low cost IMUs. I also evoke the tools I learnt to use and raise some perspectives I didn't achieve during the internship.

Contents

List of Figures

List of acronyms and abbreviations

aCOLOR : Autonomous and Collaborative Offshore Robotics
AUV : Autonomous Underwater Vehicles
DVL : Doppler Velocity Log
DOF : Degree Of Freedom
9DOF IMU : 9 degrees of freedom : acceleration in the three axis (x,y,z), the three angular rotations around the three axis and a three axis magnetometer
ESC : Electronic Speed Control
GUI : Graphical User Interface
GPS : Global Positioning System
IMU : Inertial Measurement Units
ROS : Robot Operating System
ROV : Remotely Operated underwater Vehicles
TAMK : Tampere University of Applied Science
TUT : Tampere University of Technology
USBL : Ultra-short baseline
USV : Unmanned Surface Vessel

Part I
Presentation and goals of the internship

1 Introduction

Autonomous robotic systems become more important in many industries. They started with the automation of basic tasks to lead currently to the automation of more and more elaborate ones. These robotic systems can provide a way to execute tasks faster and more safely than any human operator. In the context of a marine area, these tasks cover water exploration to underwater mining and inspection, borderline guarding to search and rescue victim in water area. [6, 9]

1.1 Issues and goal

At ENSTA Bretagne, France, student must spend twelve weeks abroad for studying reasons (e.g. academic exchange, internship). I choose to spend these three months during the internship of my first year of master. I was applying for internship in European Union without result until M. Jaulin had sent a mail with an offer for an internship at Tampere University of Technology in Tampere, Finland. The initial goal was to create and to develop a ROS package for the BlueRov2 because the USV from the project aCOLOR discribed bellow was automated with ROS, and so at the end it will easier to link the two systems.

Jose V. Escusol, doctoral student working on the USV was interested of having students from ENSTA Bretagne for the development of the BlueRov2 because there is a video of the BlueRov2 from the ENSTA Bretagne youtube channel[4]. Unfortunately I was not aware of this video but the academic program of the year was enough to start the project.

1.2 The aCOLOR project

The Autonomous and Collaborative Offshore Robotics (aCOLOR) is a project started in January 2018 is a collaboration of Tampere University of Technology (TUT), Tampere University of Applied Science (TAMK) and Alamarin-Jet Oy, a Finnish water jet manufacturer. It is a three-year project ending in 2020 founding by the European Union. The aim of the project is to combine three autonomous vehicles, one in each elements air, underwater and surface in order to achieve complete offshore inspection with one autonomous system. Indeed the global autonomous system includes an aerial vehicle, an autonomous wired underwater vehicle and a Unmanned Surface Vessel (USV).(Figure 1)

The project began by the automation the surface vessels furnished by Alamarin-Jet Oy. This USV is equipped with multiple sensors to evolve remotely and explore surface. After one and a half year, the USV is able to achieve waypoint

Figure 1: *aCOLOR project*

navigation and a obstacle avoidance in under development. Furthermore, it possess an actively controlled directional antenna to allow hi-speed, wireless and long-range communication to a Ground Control Station. The USV is also employed as launch and recovery platform for a drone and an underwater vehicle, remotely connect to the boat by a wire. These two other systems will provide air and underwater inspections.[6, 7]

1.3 BlueRov2 in the aCOLOR project

The mission of autonomous wired underwater vehicle is to provide underwater data such as camera stream and cloud points of its near environment. The aims for this vehicle is to evolve in a ray of 500m around the USV and to dive up to 500m depth. The autonomous wired underwater vehicle purchased by the team before my internship was a complete Remotely Operated Vehicle (ROV) named BlueRov2 made by the company Bluerobotics. It is a fully equipped ROV, ready to dive. I will explain how it works in a subsequent part of this report. Yet, this ROV doesn't fit the specification because its maximum depth is 300m and the tether linking him to the USV is 100m length. Hence a new frame will be developed for the ROV and the tether will be replace by a optical fiber one of 500m length.

1.4 The team

This project require a large specter of competences and many team were and will be involved to build up the project at the horizon 2020. The head of the project is Dr. Jussi Aaltonen and Prof. Kari T. Koskinen for administration and project management. The main workers for the boat and the global organisation are doctoral student Jose Villa Escusol from TUT and Sauli Virta from Alamarin-Jet Oy. There were also research assistant Samuli Niemi and master student

Aleksi Kuusisto working for the drone.

1.4.1 The BlueRov2 team

During this summer, from June to September, aCOLOR project hired four french student for working on the underwater part of the project :

- Gauthier Bussy from CESI Pau, France Mechanical engineer student
- Arnaud Salles-Thomas from CESI Pau, France, Mechanical engineer student
- Cyril Cotsaftis from ENSTA Bretagne, France, Robotic engineer student
- I, Nathan Fourniol from ENSTA Bretagne, France, Robotic engineer student.

1.5 Work organization

Gauthier, Cyril and I started the internship at TUT in Tampere on June 5 and we finished on August 30 while Arnaud joined us on August 11 and he staid until October. According to our field of study and our competence, Gauthier worked on a launch and recovery system for the ROV to deploy it in the water from the USV. Arnaud designed a new frame for the ROV while Cyril and I worked on the ROS package and control laws to automate the BlueRov2. Cyril built command laws and I took charge of linking sensors, ROV and ROS. Indeed I use to work more closely with Cyril than the others.

Part II
ROS package and first command laws

2 ROS implementation

2.1 Teamwork method and tools

Because there were a lot of people involved in the project during this summer Jose V. Escusol started a weekly meeting on monday or on friday with us, the four french people. Each one presented the work he has done the past week and what he planned for the week after.

Also, during the week, with Cyril we often made list of what we had to do to have goal and to stay motivated. We also reported on this list all the bug we encountered to not forget them and so to solve them as fast as possible. To do so we had a paper that we filled all along the internship.

2.1.1 Git and Github

The aCOLOR is an open-source research project, even Jose V. Escusol used Matlab from Mathwork to deal with the USV. So we were encouraged to use only open-source solutions.

Because Cyril and I worked on the same ROS package, I suggested to use git : a version-control of our code. It allowed us to easely follow the modifications and to share between us the code we have done thanks to github. It was also a security to not loose our work in case of trouble on our computer. The repository is accessible at `https://github.com/cyrilcotsaftis/bluerov_ros_playground`. We worked only on the master branch wich means if someone stored bug files, we would not have a clean branch with source code without bug. For our project size it was almost enough but for bit bigger project it could be really useful to use the branch system of git to always have a branch with source code without computer bugs.

2.1.2 Gitbook

This french open-source website`https://www.gitbook.com/`, which also exists as software, was useful to write the documentation of our code. This documentation is stored online at `https://app.gitbook.com/@acolor-bluerov2/s/acolor-bluerov2/`. It is easy to use and the result looks really nice without tons of efforts.

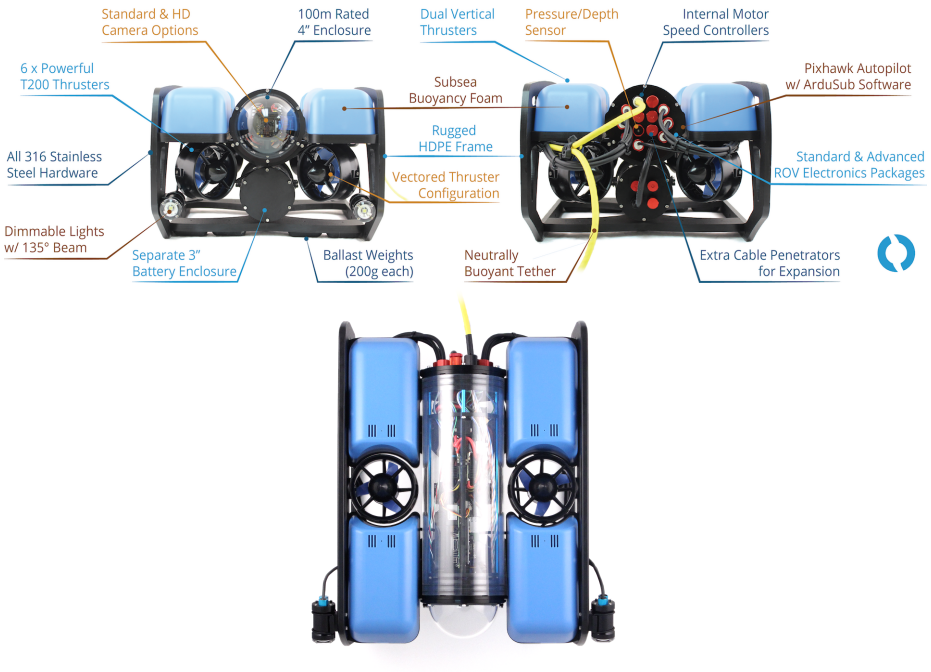


Figure 2: BlueROV2

2.1.3 ROS

"ROS is a robotic middelware, a collection of software framework for robot software development"[2]. This tool provide a great abstraction for client server program. With ROS there is node, corresponding to the execution of the code and topics, a place that receive messages published from node. The main pros is that nodes can subscribe to topics to listen messages and publish on topics to share data with other nodes.

2.2 ROS package

2.2.1 BlueRov2 component

BlueRov2 is a 45x438x254mm underwater robot sold by Bluerobotics. It belongs to the ROV category wich means it is remotly controlled. It has six thrusters, two horizontal for roll and altitude and four vertical. This four thrusters are vector positioned to allow four degrees of freedom : lateral, forward, backward, rotation in the horizontal plane. To command this thruster, the BlueRov is linked to a topside computer by a 100m tether.

MAVros	mavgen
Written in C++	Library for C,C++ and python
ROS package ready to use	Need to write our codes
Maintained by the comunity	Need to be maintained by the project
Not specific to the BlueROV2	Adapted to the project

Table 1: MAVros versus mavgen

2.2.2 MAVlink

To communicate, the ROV used MAVlink witch is a lightweight messaging protocol design for drones and moving robots[10]. To deal with this message, the ROV runs Ardusub, a branch of ardupilot project for underwater vehicles on a Pixhawk, a hardware electronic card to easely connect sensors and pwm motors with MAVlink. On the topside computer, the software QGroundControl is made for handeling MAVlink message, configure Ardusub on the ROV and drive it with a gamepad.

The issue to make automomous mission with the ROV was to handle MAVlink messages and transform it into ROS messages. To do so their is the mavgen library to handle it. To use it there are two options : the MAVros package a more or less ready to use solution or to use the mavgen library to write our own scripts.

With Cyril we choose to write our own program in python using mavgen library for Python named PyMavlink. The main reasons for this choice was because I found some a package on github, from Patrick José Pereira, a software development ingineer from Bluerobotics who started to write a ROS package to deal with the BlueRov[8]. So we started to improve and make modifications to have the architecture that we discussed and wanted.

2.2.3 ROS architecture

The ROS architecture we decided to build is organised around four different namespaces. The main one is the workspace BlueRov2 because it is the bridge between ROS and MAVlink. It handles incoming messages from MAVlink and publish them on ROS topics. At the same time, ROS messages published on topics are converted and sended with MAVlink to the ROV.

After this bridge, we build a command namespace where all conrollers, depth, heading, velocity and the gamepad publish their command. Next the commander node subscribe to those topics and publish on thruster channel command commands from the controllers.

To deal with parameters of our scripts, we added a Settings namespace to modify in real time coefficients and target for controllers.

At the end we have the connection as we have on the figure 4.

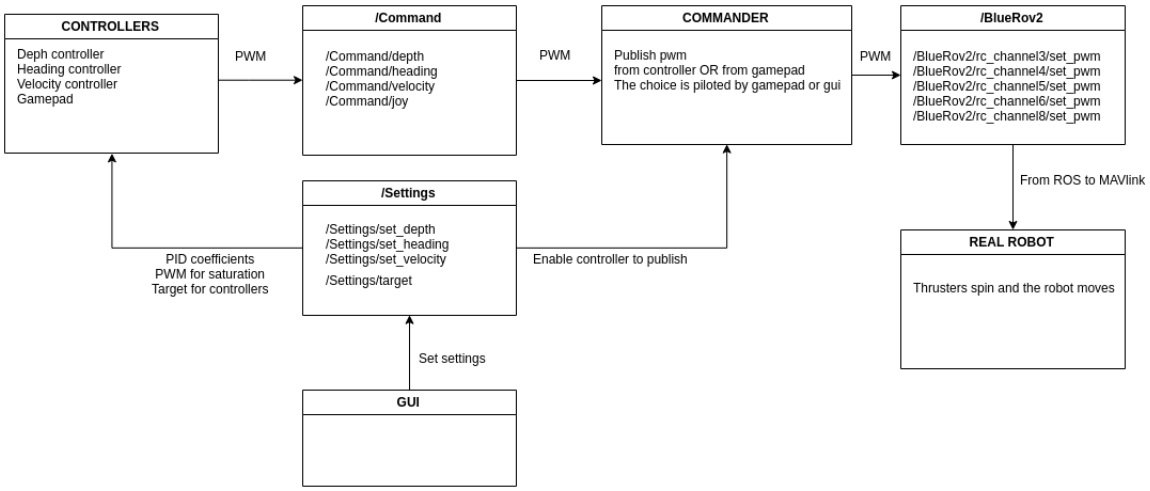


Figure 3: ROS data flow and architecture

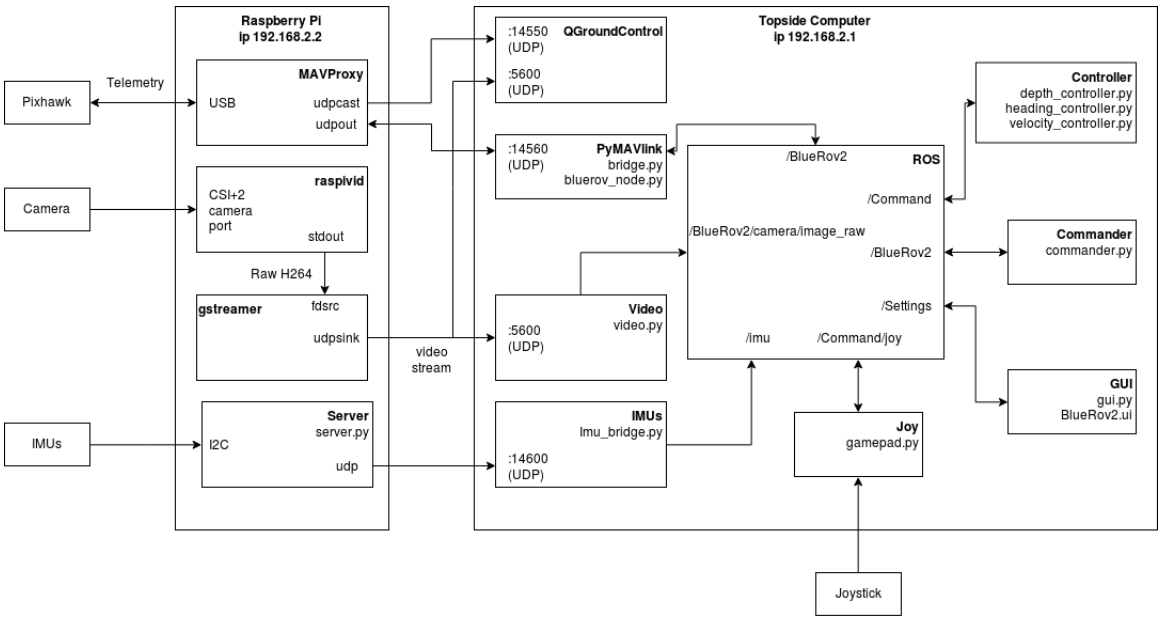


Figure 4: Connection model

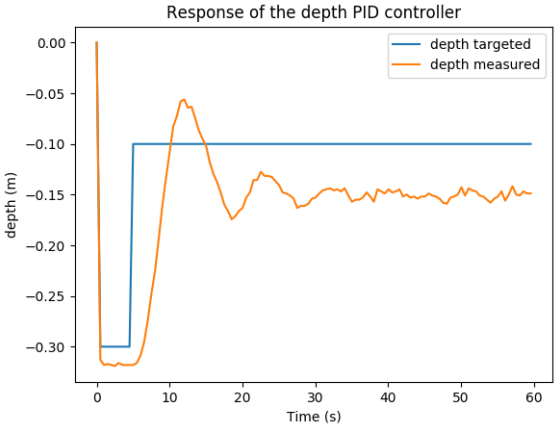


Figure 5: Response of the depth PID controller

2.3 Command laws

2.3.1 Depth control

The Bluerov is equiped with a Bar30 sensor from Blurobotics. It a pressure sensor up to 300m depth with 0.2mbar resolution. So to estimate current depth the sensor gets the absolute pressure and computes it with hydrostatic equation and the pressure at surface.

$$depth = -(p - p_0)/(\rho * g)$$

Designation	Meaning	Value
$depth$	current depth	m
p	pressure measured	Pa
p_0	surface pressure	99 000Pa
ρ	water density	1000 kg/m^3
g	gravitational acceleration	9.81 kg/m^2

To reach the wanted value of depth, Cyril implemented a PID controller (Fig 5)and a saturation method because thrusters are controlled by a pwm in range of 1100 to 1900. We can observe that it still have an error of five centimeters. Its mainly due to the facts that the command sended by the controller at the end is too small to pull up the robot.

2.3.2 Heading command

The Bluerov is also equiped with IMU with 9 DOF. Angular acceleration and magnetometer orientation are compute in the Pixhawk to return the orientation quaternion of the ROV. With this orientation, we designed a PD controller and again a saturation method. Because quaternion are difficult to understand for

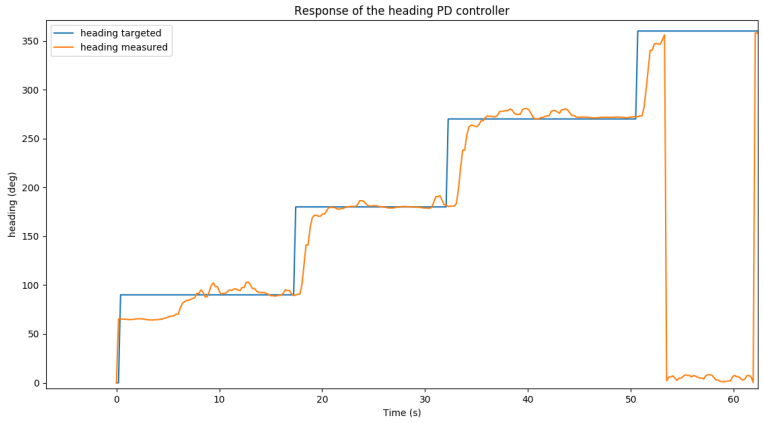


Figure 6: Response of the heading PD controller

a human, the heading desired in a angle value in 0-360 degree range, where 0 is magnetic North, 180 South (Fig 6). In this graph, we saw that the heading is reach even for the 360 degrees command because it's the same as a 0 degree command.

2.3.3 Velocity command

After that, during the internship, with Cyril, we were asked to try to estimate velocity only with two 20$ adafruit 9DOF IMUs. We read a lot that acceleration integration to get velocity with only low cost IMUs is not accurate because of the continous integration of the errors. However, Jose Villa Escusol wanted us to do as much as possible with this stuff and it was a great experience even we couldn't get accurate velocity from those two IMUs.

The first step was to set IMU in the ROV and establish a connection with the topside computer. This two IMU were linked with I2C at the internal Rasberry Pi, used in the ROV to stream video to topside computer and now it also sends acceleration datas to the topside computer.

Next, this IMUs needed calibration. To do so we used FreeIMU calibration gui, a calibration software made by F. Varesano during his PhD at UNITO in Italia that I modified to work with ROS topics instead of serial Arduino link (Fig 7). After that, we had a calibration value with the following equation :

$$CalibratedValue = (MeasuredValue - offset)/scale$$

Where the offset and scale is set to have all points in a sphere with a radius of 1, centered on 0.

After that the signal was better but required to be low pass filtered. On figure 8 we can see in red raw values for acceleration along the X axis and

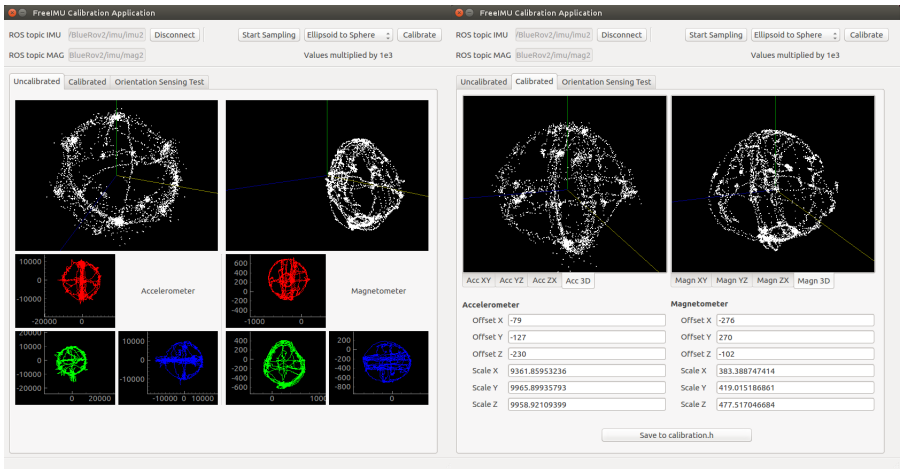


Figure 7: FreeIMU calibration tools

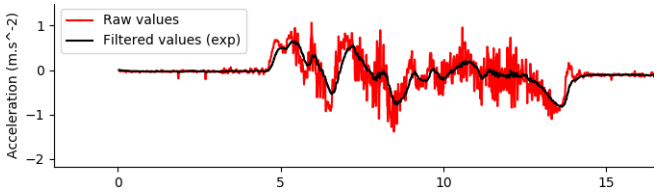


Figure 8: IMUs experimental values

in black the filtered signal with an exponential filter also known as IIR filter (infinite impulse response filter). Its equation is :

$$y[n] = a*x[n] + (1-a)*y[n-1]$$

Where y is the output and x the measured vector, a is the smoothing value.

Finally while the acceleration is calibrated and filtered, we started to integrate then using three method : the euler and simpson method and Cyril tries by integrating and merge datas with a Kalman filter. To merge the two IMUs we did the mean of the two signals. In all cases, they were to much drift in the velocity even we can see some movement forward and backward according to the variation of the curves (Fig 9). The linear parts correspond to a not moving robot.

According to litterature , to improve the value of velocity we must add a non inetial sensor such as doppler sensor or an ultra short baseline (USBL) system, a local positionning system for underwater materials. Then we can merge this new value with acceleration ones in a Kalman filter to have a better estimation of the position or the velocity of our ROV. But this two systems were to expensive for the project yet.

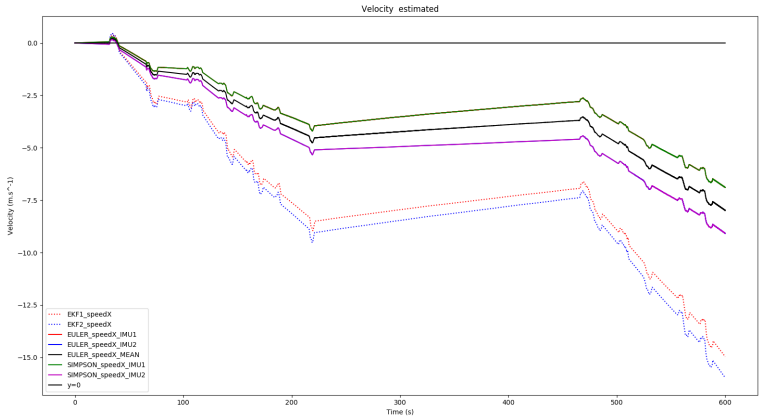


Figure 9: Comparison between several method to integrate acceleration signals

2.4 GUI

With this archicture, it was not so easy to change parameters for controllers during test and while the ROV was working. I choose to devellop a small Qt interface to change controller parameters, deal with autonomous or manual command, set target for each controllers and have an overwiev of the output of the controllers (Fig 10). At the end, I added option to record easily datas with rosbag, a tool provided by ROS to record message and topics .

2.5 Tests

In order to test my codes, I use a lot the simulation of the ROV made by the Ardupilot project (Fig 11). This step was usefull to solve communication issues in the ROS architecture and the conversion between ROS and MAVlink message.

Futhermore, in the lab of the university, I have access to a water tank to run test with the robot in water. I could test the depth and heading control and tune the parameters for the controllers.

At the end, the university in Tampere was not so far from a lake. With Cyril we used it as a great place to play and test our codes. We also test a mission where the ROV had to do a square.

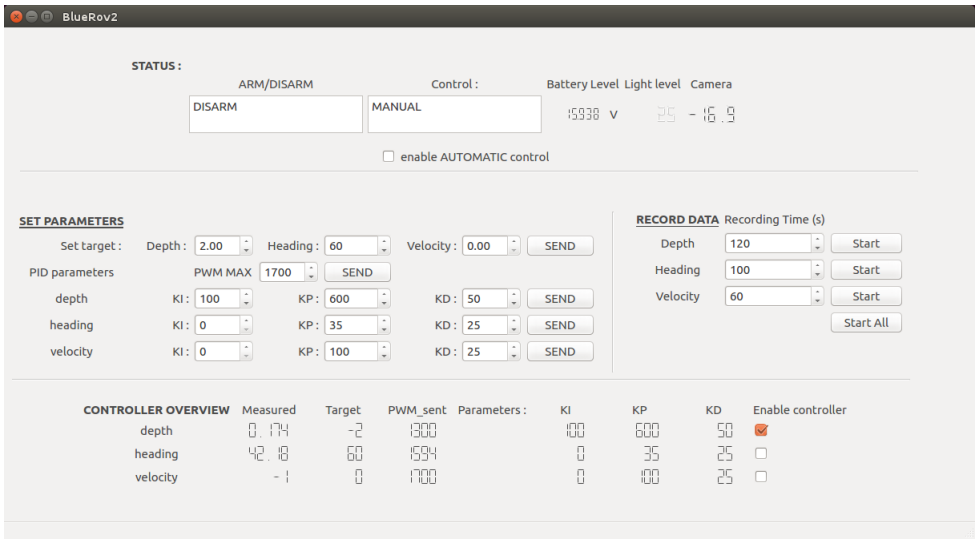


Figure 10: GUI

Figure 11: SITL and lake test

Part III
Analysis of the internship

3 Input of the internship

3.1 Knowledge aquired

In this intership, I really learnt how to work in team. I came up with open source tools as github and gitbook to share and combine my work with Cyril one. With this tools it was easy to follow the evolution of the ROS package and to track bug.

Futhermore, I really improve myself in programing and program organisation. I was behind all scripts trying to follow the same organisation in code files to make them better understandable, maintenable and sustainable.

3.2 What I could have done better

There is a lot of things that we could have done better. First it would be better if we had set up a tool to schedule our work instead of a floating paper handly written. There are plenty of one as Framaboard from Framasoft to quote an opensource solution. Secondly, I still wonder if the choice to not use MAVros is still a good option because it seems more maintenable because of a strong community. The last things is about IMUs and inertial navigation. It could have been great if we tried to better quantize errors and to tried other method then the mean to merge datas for the 2 IMUs.

4 Conclusion

This traineeship has been a very interesting and instructive experience on research Laboratory in Tampere University of Technology. It was great time of working on an exiting project. I started to understand how the BlueRov2 worked as it was shipped. Next I started develloping a ROS package to handle communication with the BlueRov2. After that with Cyril, we build depth and heading controller. In addition we moved on trying to estimate velocity with low cost IMU witthout success. However it was a useful to improve my skill in programming and codes organisation as well as I learnt a lot of how to use git and opensource tools to build a productive environment.

References

[1] Fusion of gps and redundant imu data for attitude estimation, 08 2012.

[2] Inertial navigation in marine robotics, 2019.

[3] Jared B. Bancroft and Gerard Lachapelle. Data fusion algorithms for multiple inertial measurement units. *Sensors*, 11:6771–6798, 06 2011.

[4] ENSTA Bretagne. Bluerov2 simple autonomous mission, 03 2018.

[5] edmallon. Tutorial: How to calibrate a compass (and accelerometer) with arduino, 02 2019.

[6] Jose Villa Escusol. Air, water surface and underwater inspections with just one autonomous robotic system, 01 2018.

[7] Jose Villa Escusol. acolor achieves first year goals of the project with the unmanned surface vessel as a main topic., 01 2019.

[8] Patrick Jose Pereira. bluerov ros playground, 01 2018.

[9] Kari T. Koskinen. Autonomous and collaborative offshore robotics (acolor), 01 2018.

[10] MAVLink. Mavlink developer guide, 2009.

[11] Matthew Parrilla. Inertial navigation in marine robotics, 2019.

[12] Fabio Varesano. Freeimu magnetometer and accelerometer calibration gui: Alpha version out, 2012.

ENSTA Bretagne

RAPPORT D'EVALUATION
ASSESSMENT REPORT

Merci de retourner ce rapport par courrier ou par voie électronique en fin du stage à :
At the end of the internship, please return this report via mail or email to:

ENSTA Bretagne – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE
☎ 00.33 (0) 2.98.34.87.70 / stages@ensta-bretagne.fr

I - ORGANISME / *HOST ORGANISATION*

NOM / *Name* Tampere University

Adresse / *Address* Korkeakoulunkatu 6, 33720, Tampere (Finland)

Tél / *Phone (including country and area code)* +358 40 849 0522

Nom du superviseur / *Name of internship supervisor*
Jussi Aaltonen
Fonction / *Function* Research Manager

Adresse e-mail / *E-mail address* jussi.aaltonen@tuni.fi

Nom du stagiaire accueilli / *Name of intern* Nathan FOURNIOL

II - EVALUATION / *ASSESSMENT*

Veuillez attribuer une note, en encerclant la lettre appropriée, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
*Please attribute a mark from **A (excellent)** to **F (very weak).***

MISSION / *TASK*

- La mission de départ a-t-elle été remplie ?
  Was the initial contract carried out to your satisfaction? (A) B C D E F

- Manquait-il au stagiaire des connaissances ? ☐ oui/*yes* ☒ non/*no*
  Was the intern lacking skills?

  Si oui, lesquelles ? / *If so, which skills?* _____

ESPRIT D'EQUIPE / *TEAM SPIRIT*

- Le stagiaire s'est-il bien intégré dans l'organisme d'accueil (disponible, sérieux, s'est adapté au travail en groupe) / *Did the intern easily integrate the host organisation? (flexible, conscientious, adapted to team work)*

  (A) B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* The intern integrated perfectly from the beginning of the internship, having a great team work in the Mechatronics Research Group (MRG).

COMPORTEMENT AU TRAVAIL / *BEHAVIOUR TOWARDS WORK*

Le comportement du stagiaire était-il conforme à vos attentes (Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances) ?
Did the intern live up to expectations? (Punctual, methodical, responsive to management instructions, attentive to quality, concerned with acquiring new skills)?

(A) B C D E F

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

INITIATIVE – AUTONOMIE / *INITIATIVE – AUTONOMY*

Le stagiaire s'est –il rapidement adapté à de nouvelles situations ? A B C D E F
(Proposition de solutions aux problèmes rencontrés, autonomie dans le travail, etc.)

Did the intern adapt well to new situations? (A) B C D E F
(eg. suggested solutions to problems encountered, demonstrated autonomy in his/her job, etc.)

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* Intern had great autonomy during the internship, working in different tasks related to an Autonomous Underwater Vehicle.

CULTUREL – COMMUNICATION / *CULTURAL – COMMUNICATION*

Le stagiaire était-il ouvert, d'une manière générale, à la communication ? (A) B C D E F
Was the intern open to listening and expressing himself /herself?

Souhaitez-vous nous faire part d'observations ou suggestions ? / *If you wish to comment or make a suggestion, please do so here* _____

_____

OPINION GLOBALE / *OVERALL ASSESSMENT*

❖ La valeur technique du stagiaire était : (A) B C D E F
Please evaluate the technical skills of the intern:

III - PARTENARIAT FUTUR / *FUTURE PARTNERSHIP*

❖ Etes-vous prêt à accueillir un autre stagiaire l'an prochain ?

Would you be willing to host another intern next year? [X] oui/*yes* [ ] non/*no*

Fait à _____, le _____
In Tampere, on 26.08.2019

JUSSI AALTONEN

Signature Entreprise _____ Signature stagiaire
Company stamp _____ *Intern's signature*

Tampereen yliopisto Universitas Tamperensis Tampere University

Merci pour votre coopération
We thank you very much for your cooperation