



Second year internship report
Autonomous and Collaborative Offshore Robotics (aColor)
ROS implementation on the BlueROV2



ENSTA Bretagne
 2 rue F. Verny
 29806 Brest Cedex 9, France

Cotsaftis, Cyril, cyril.cotsaftis@ensta-bretagne.org
 Robotics student



Tampere University Foundation
 Korkeakoulunkatu 6
 33720
 TAMPERE (Finland)

Jose Villa Escusol
 Doctoral student at Mechatronics Research

Jussi Aaltonen
 Research Manager (D.Sc. Tech.)
 Mechatronics Research Group

Résumé

Dans le cadre de mes études en robotique à l'ENSTA Bretagne, j'ai eu l'opportunité de partir en Finlande à Tampere University afin d'y réaliser un stage d'assistant ingénieur d'une durée de 3 mois. J'ai travaillé sur un projet académique et international : aColor. L'objectif du projet est la réalisation d'un système autonome alliant 3 sous-systèmes évoluant en milieu terrestre, aérien et sous-marin. Les applications sont multiples, réalisation de mapping 3D d'environnement inconnu, mission de sauvetage, inspections d'installations sous-marines pour de l'entretien. Ma mission au sein du projet est l'implémentation d'une architecture ROS permettant l'automatisation du robot sous-marin BlueROV2. Ce stage fut l'occasion de valider mon cursus nécessaire à l'obtention de mon diplôme d'ingénieur mais également une opportunité de mettre en application mes connaissances en robotique. J'ai pu par ailleurs améliorer mes compétences en anglais, aussi bien dans un cadre professionnel que quotidien. Le projet aColor est un projet académique ne disposant pas de moyens financiers très importants, il était donc essentiel de faire preuve d'ingéniosité pour trouver des solutions innovantes à moindre coût. De plus, la communication est essentielle pour mener à bien le projet. Elle permet la coordination au sein de l'équipe mais est aussi un moyen de partage d'idées et de connaissances.

Abstract

Thanks to my studies in robotics at ENSTA Bretagne I was able to do a 3-month internship as engineer assistant at Tampere University in Finland. I worked on the aColor project. The goal of this project is to craft an autonomous system composed of 3 sub-systems: a boat, a drone and a submarine. The system will then be used for 3D-mapping, search and rescue missions and underwater inspections. The field of applications is wide. My mission was the implementation of a ROS-architecture in the BlueRov2, which enable its automation. During my internship, I improved my tongue skills (written and spoken), I also applied my robotics knowledge to a practical case. The teamwork was essential to make progress, by sharing ideas during brainstorming sessions and solving problems together. Moreover, being able to work in autonomy has been beneficial.

Acronyms

ROV : Remotely Operated underwater Vehicles

AUV : Autonomous Underwater Vehicles

ROS : Robot Operating System

aColor : Autonomous and Collaborative Offshore Robotics

IMU : Inertial Measurement Units

DVL : Doppler Velocity Log

GPS : Global Positioning System

USBL : Ultra-short baseline

ESC : Electronic Speed Control

DOF : Degree Of Freedom

Contents

| | |
|---|----|
| Résumé | 2 |
| Abstract | 3 |
| Acronyms..... | 4 |
| I. Presentation and goals of the internship | 7 |
| 1. Tampere University | 7 |
| 2. aColor Project | 7 |
| 3. The team..... | 8 |
| 4. The internship..... | 9 |
| 5. Tools used to work | 9 |
| II. About the BlueROV2..... | 10 |
| 1. Physical architecture | 10 |
| 2. Connection | 11 |
| III. What have been done | 12 |
| 1. ROS Architecture | 12 |
| i. Controllers and Commander | 13 |
| ii. ROS-MAVlink bridge | 14 |
| iii. IMU server | 14 |
| 2. Command laws | 15 |
| i. Heading..... | 15 |
| ii. Depth | 16 |
| iii. Velocity..... | 17 |
| 3. IMU and velocity estimation | 17 |
| i. About IMUs..... | 17 |
| ii. Why 2 additional IMUs | 17 |
| iii. Raw IMU data computation | 17 |
| 4. Graphical User Interface (GUI) | 19 |
| i. Status | 20 |
| ii. Set parameters | 21 |
| iii. Record data | 21 |
| iv. Controller overview | 22 |
| 5. Simulation and tests | 23 |

| | |
|----------------------|----|
| i. Simulation..... | 23 |
| ii. Tests..... | 23 |
| Conclusion | 24 |
| List of Figures..... | 25 |
| Bibliography..... | 26 |

I. Presentation and goals of the internship

1. Tampere University



Figure 1 : Aerial view of the university

Tampere University is Finland's second-largest university in engineering sciences. It is in Hervanta, a suburb of Tampere. There are 5 faculties in Biomedical Sciences, Business, Electrical Engineering, Natural Sciences and Engineering Sciences. Each Faculty is composed of several department.

During my internship, I was working in Konetalo from the department of Mechanical Engineering and Industrial Systems. A lot of tools are available to work in good conditions, a big library, 3D printers, IT places, laboratories... The University is also open to the world and numerous international projects as been started, as aColor project.

2. aColor Project

Autonomous robotic systems may be the future of many industries. They are safe, fast and do not require any human interaction. One of its application field is offshore inspections, which is currently labour intensive.

The project "Autonomous and Collaborative Offshore Robotics" has started in January 2018 and will last for 3 years. It is a collaboration between Tampere University of Technology (TUT), Tampere University of Applied Science (TAMK) and Alamarin-Jet Oy, a Finnish water jet manufacturer. The project is founding by the European Union. The objective of the project is to develop autonomous and collaborative capabilities between three subsystems: Unmanned Surface Vessel (USV), Unmanned Aerial Vehicle (UAV) and Autonomous Underwater Vehicle (AUV). The system would then be able to complete missions such as underwater surveys (mapping/inspections), recovering and launching fixed wing UAV from moving surface vessel and search and rescue (SAR) operations.



Figure 2 : aColor subsystems

The first year of the project was dedicated to the development of the vessel, it is now fully autonomous and able to follow waypoints. The vessel is used as a launch and recovery platform for the UAV and the AUV. Yet it remains an academic project managed by a small crew, so a key point of this project it to be ingenious in order to cut prices.

3. The team

Prof. Kari T. Koskinen, TUT
 Dr. Jussi Aaltonen, TUT
 Prof. Moncef Gabbouj, TUT
 Prof. Mikko Valkama, TUT
 Doctoral Student Jose Villa Escusol, TUT
 Research Assistant Samuli Niemi, TUT
 Master Student Aleksu Kuusisto, TUT
 Master Student Gauthier Bussy, CESI Pau
 Master Student Cyril Cotsaftis, ENSTA Bretagne
 Master Student Nathan Fourniol, ENSTA Bretagne
 Master Student Arnaud Salles-Thomas, CESI Pau

Jussi is the manager of the project, he decided and approve our suggestions for the project. He also gave us some valuable advice on how to get further when we encounter an issue.

Jose works on the automation of the boat. He has been our tutor for the internship and was always there to answer our questions or to help us in any way.

Samuli works on the drone.

Gauthier designed a launch and recovery structure for the BlueROV2.

Arnaud designed a new 3D-structure for the BlueROV2 at aim to install more equipment such as sensors and cameras. The point here is to get a nice view around the robot for mapping applications.

Nathan and I worked on the ROS implementation for the BlueROV2.

4. The internship

The activity at Tampere University slows down during summer due to vacation, so taking interns for the summer was a way to help with that. For me it was the occasion to study abroad, to improve my tongue skills, to travel but also to apply my theoretical knowledge on a practical project.

For my internship, I have been assigned to work on the underwater vehicle. The team bought a Remotely Operated Vehicle (ROV) from BlueRobotics called BlueROV2. The final goal is to make it autonomous as the other subsystems. For that, Jose asked to implement a ROS architecture in the robot in order to command it, without being too specific on what was needed.

When we arrived, the robot was brand-new and so nothing was implemented yet. It only had a manual mode, that allows to pilot the robot with a joystick and to make sure the installation was done properly and that it works fine.

The first week of the internship was dedicated to research and documentation. Our initial goal was to understand how the robot works, find a way to simulate it and then think about a ROS structure.

We also did the topside computer setup, installing ubuntu, ROS, and libraries.

5. Tools used to work

In order to work in good conditions and to make fast progress we decided to use some useful tools.

We made a Github to share codes easily, as well as a gitbook. The gitbook is a user guide that explain how the robot work and detailed the work that we have done during this summer internship. That way, it will be easier for the following workers to keep up.

Moreover, we had weekly meetings with the team to share progress and discuss new goals and ideas.

When working on project, communication is key to get things done. Being aware on what others are doing and how, is helpful for the advancement of my own tasks.

II. About the BlueROV2

“The *BlueROV2* is the most capable, flexible, and affordable ROV on the market. The 6-thruster vectored configuration, coupled with strong static stability, provides a vehicle that is smooth and stable, yet highly maneuverable. The *BlueROV2* provides the capabilities of a high-end mini-ROV at the price of the most basic commercial ROVs.”

1. Physical architecture

The BlueRov2 is a small Remotely Operated Vehicle (ROV) made and sold by the company Bluerobotics. The BlueRov2 is equipped with several sensors and actuators:

- 6 Thrusters T200
- 6 Basic Esc
- 1 Bar30 pressure sensor
- 2 Leak sensors
- 4 Lumen Subsea Lights
- 1 Raspberry Pi 3B, used as a companion computer
- 1 camera
- 1 servo for the camera tilt
- 1 PixHawk (Px4-v2) with internal 9 DOF IMU
- 2 Adafruit LSM9DS1 9DOF
- 1 Fathom ROV tether

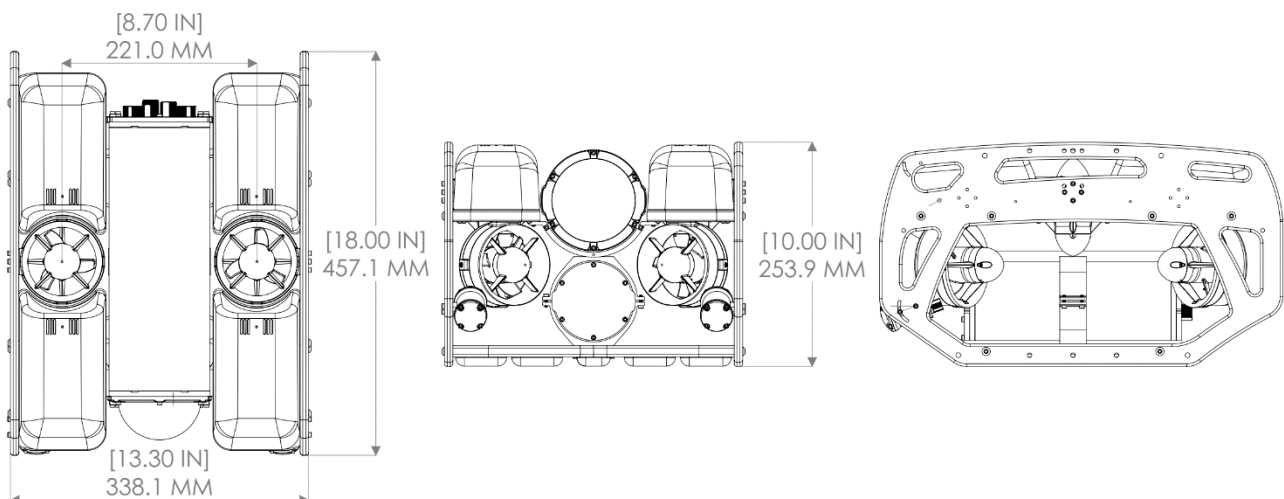


Figure 3 : 2D Drawings

The small size of the BlueROV2 and its high manoeuvrability is perfect to achieve exploration missions in tiny places. Also, its wide variety of sensors make the automation easier.

2. Connection

According to the ArduSub documentation, the BlueRov2 requires a tether to be linked to a topside computer. Figure 3 shows a diagram of how it is connected to a topside controller with the ROS implementation.

QGroundControl can be used to set up various parameters and to pilot manually the robot while having the view of the camera.

ROS is used to compute command laws from sensors data. These commands are PWM values for the motors inside the BlueROV2.

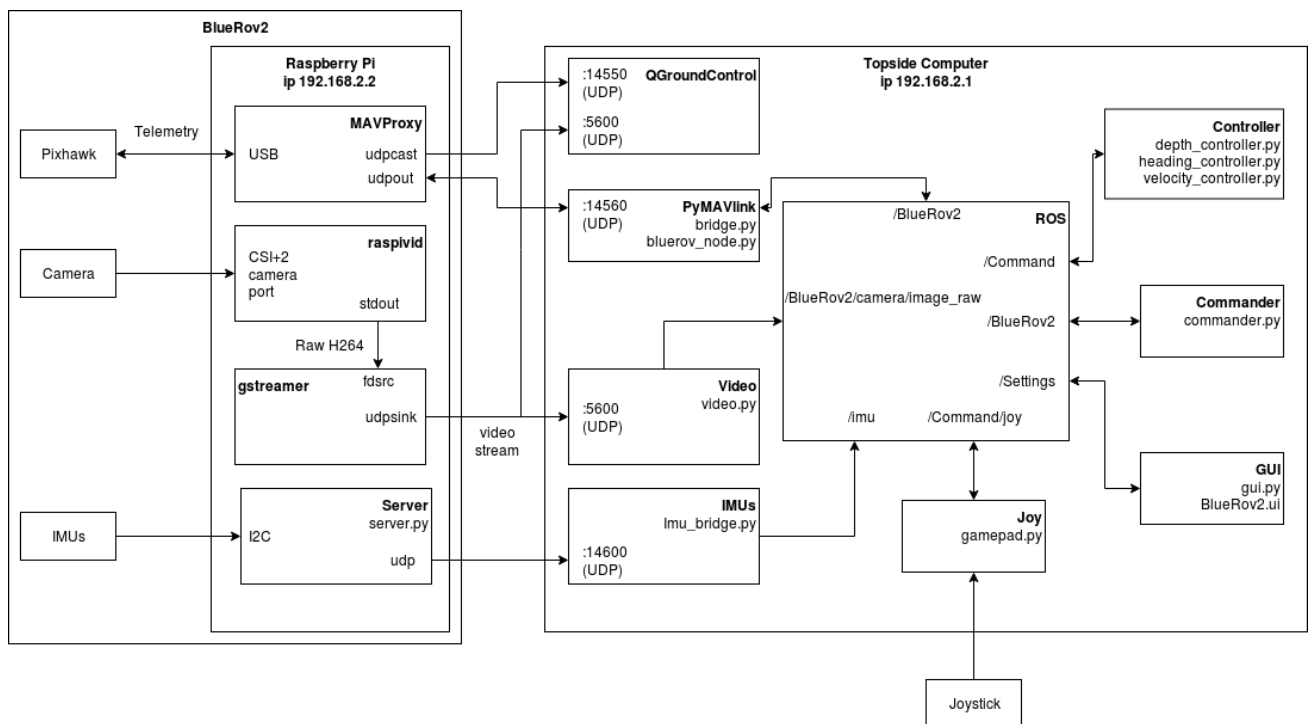


Figure 3 : Connection model

On the left side, we see the structure of the BlueRov2. It has sensors and a companion computer (Raspberry PI). The companion computer is an intermediate between BlueRov2 sensors and ROS. Command values from ROS are sent via MAVlink to the raspberry, and then to the Pixhawk. The Pixhawk is connected to ESC that controls motors.

III. What have been done

1. ROS Architecture

The control of the BlueROV2 works has following:

- Sensors on the robot collect valuable data of its surrounding and gives us a direct or indirect knowing of its physical state.
- These data are then sent and processed in the controller that will output a command for the actuators. The command evolves with time according to the desired state and the current one.
- The generated command is then sent to the commander. It enables a joystick mode or an automatic mode.
- The commander sent the command to the actuators.

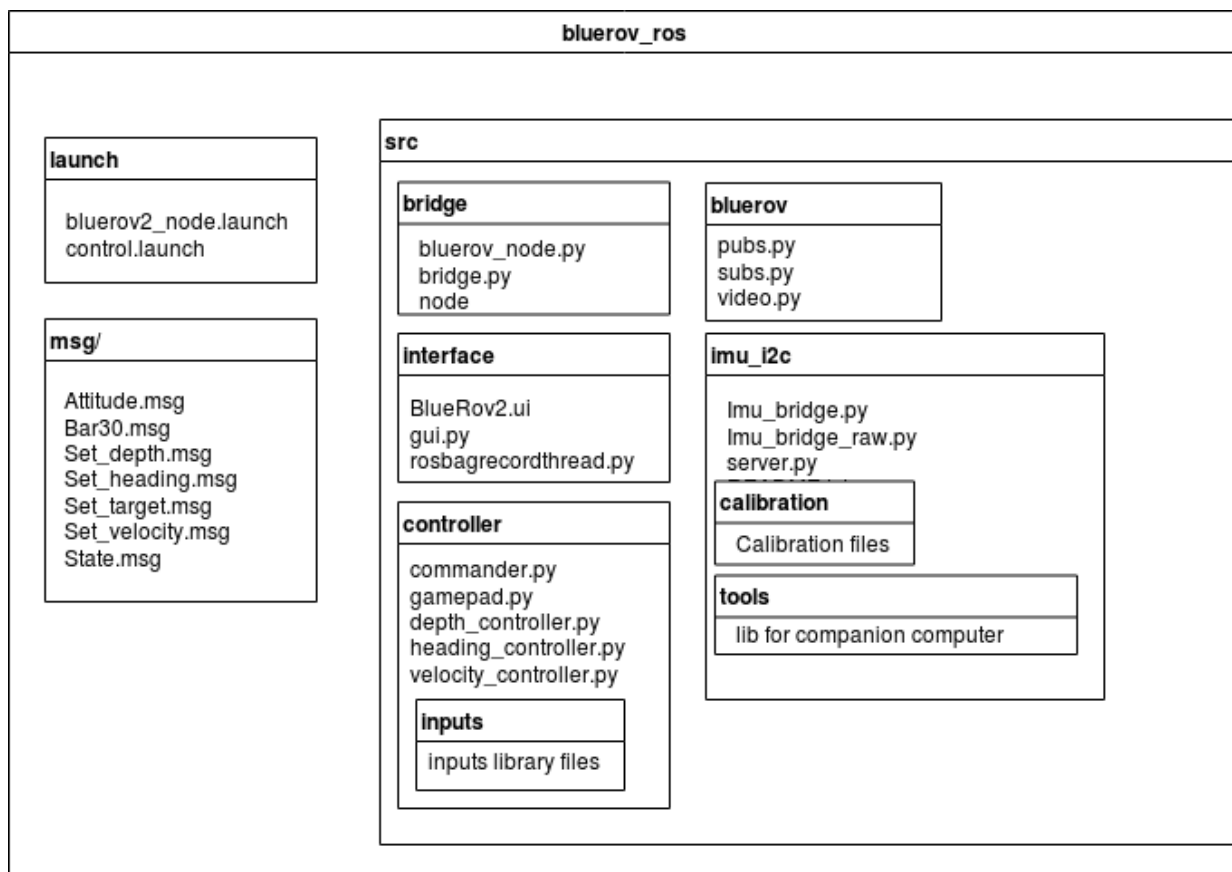


Figure 4 : ROS package structure

The launch file makes it easy to start a mission or to do tests.

i. Controllers and Commander

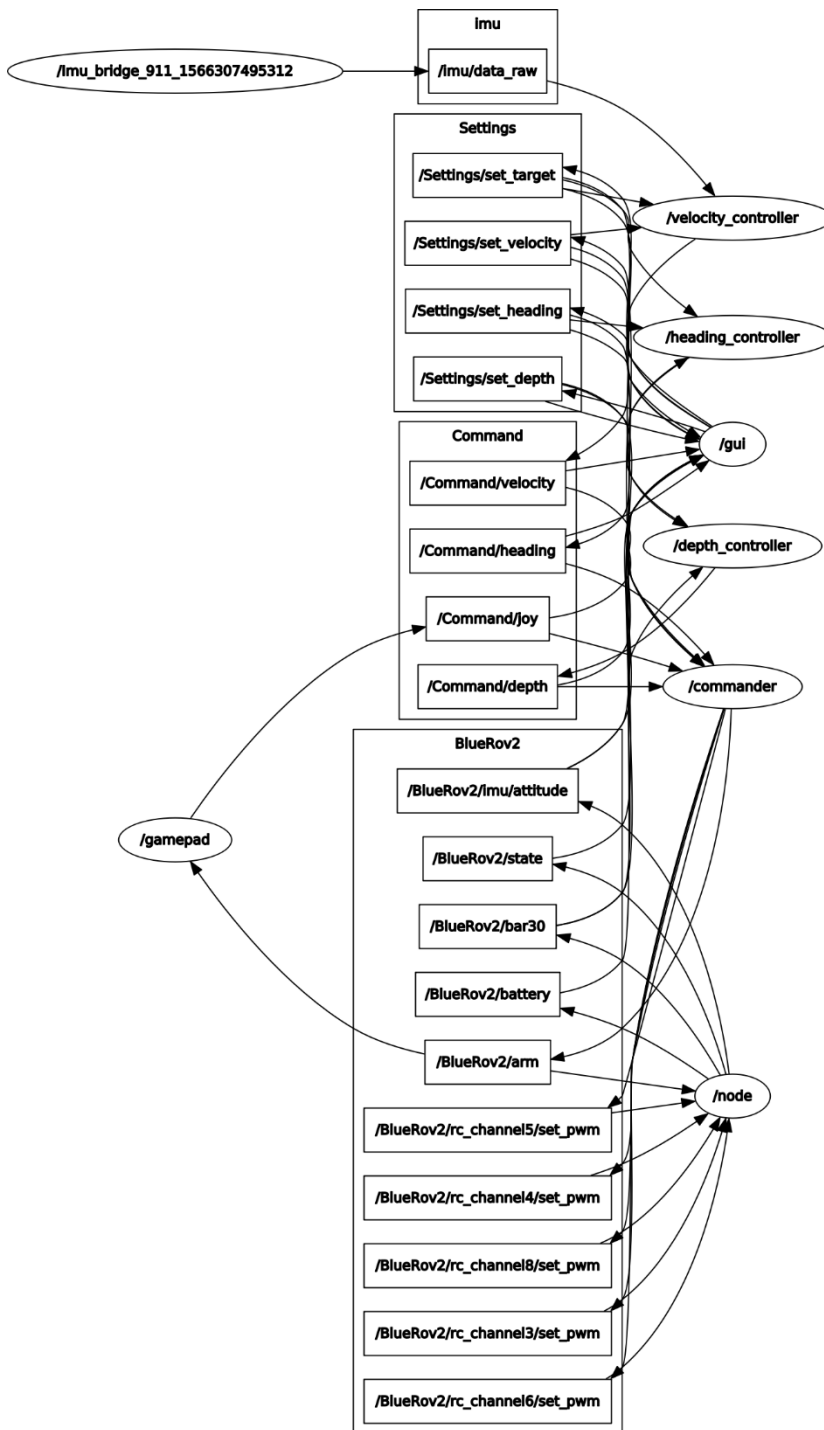


Figure 5 : Final ROS-graph

This graph appears to be confusing because of its complexity, however the structure itself really isn't. Let's use an early stage of the implementation as an example to explain how it works:

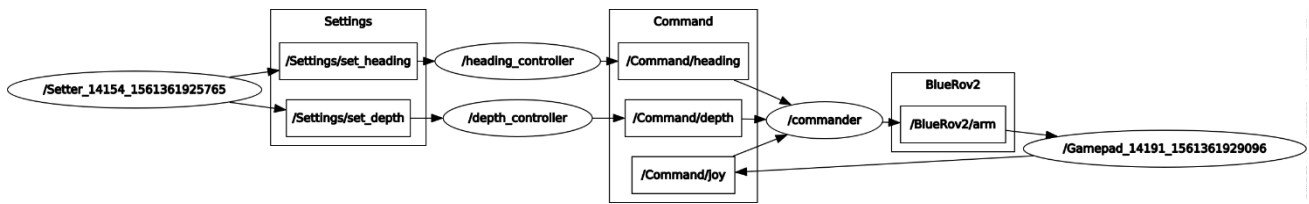


Figure 6 : ROS-graph early stage

We set desired values (manually and later with a GUI) and publish them on Settings topic. Controllers nodes are subscribed to Settings and to sensors topics (not represented on this example, BlueRov2 topics). The command value is then published on Command topic. The commander node is the only link between Command topic and BlueRov2 topic.

The core of ROS structure is the controller's nodes. There are 3 of them for depth, heading and velocity. The first step is to estimate those values.

For the depth, we used a pressure value and thanks to hydrostatics equations we can indirectly measure depth.

The heading can be estimated with an IMU.

Having a velocity measurement is far more complicated without highly expensive sensors. All we had for that was IMU's.

The second step is to compute a command law. It is explained in the section below.

The idea behind the commander, was to have a "filter" between all controllers and the robot. The commander is used to coordinate everything, even the pilot mode desired (joystick mode or autonomous mode).

There are two modes in commander: the MANUAL mode where the gamepad controls the ROV and the AUTOMATIC mode where depth, heading and velocity PWM computed by the controllers drive the ROV if there are enabled to be published.

ii. ROS-MAVlink bridge

The ROS-MAVlink bridge is a datalink between the offboard (ROS, on our computer) and the companion computer (the robot, so embedded computer)

iii. IMU server

In order to command the robot, we decided to add 2 more IMU, with high rate of acquisition. Since this rate is higher than maximal rate on ROS, additional IMUs data are sent from a server on the robot to a client on the offboard. There is finally a topic on ROS to get the value back and process it.

2. Command laws

The controllers implemented for heading, depth and velocity are PD or PID. It depends on the sensors data that are available. We didn't have the state equations of the robot, so we chose to work with a model-free control.

i. Heading

In order to control the heading, we first had to estimate the yaw. The estimation is given from angular speed of the IMU from which we compute the orientation quaternion that allow us to estimate the yaw. Since we know the yaw and its derivative, we can implement a proportional-derivative controller.

We give a desired yaw and the controller returns a PWM command for the motors. The closer we get to the desired position, the smaller the PWM value become. Moreover, this PWM command goes through a saturation function to limit thruster power.

To deal with the 2 PI modulo for angles, the difference between yaw desired and current yaw goes through a function sawtooth.

The parameters of the controller have been chosen experimentally.

$$u = \mathbf{KP} * \text{sawtooth}(\text{yaw} - \text{heading_desired}) + \mathbf{KD} * \text{yawspeed}$$

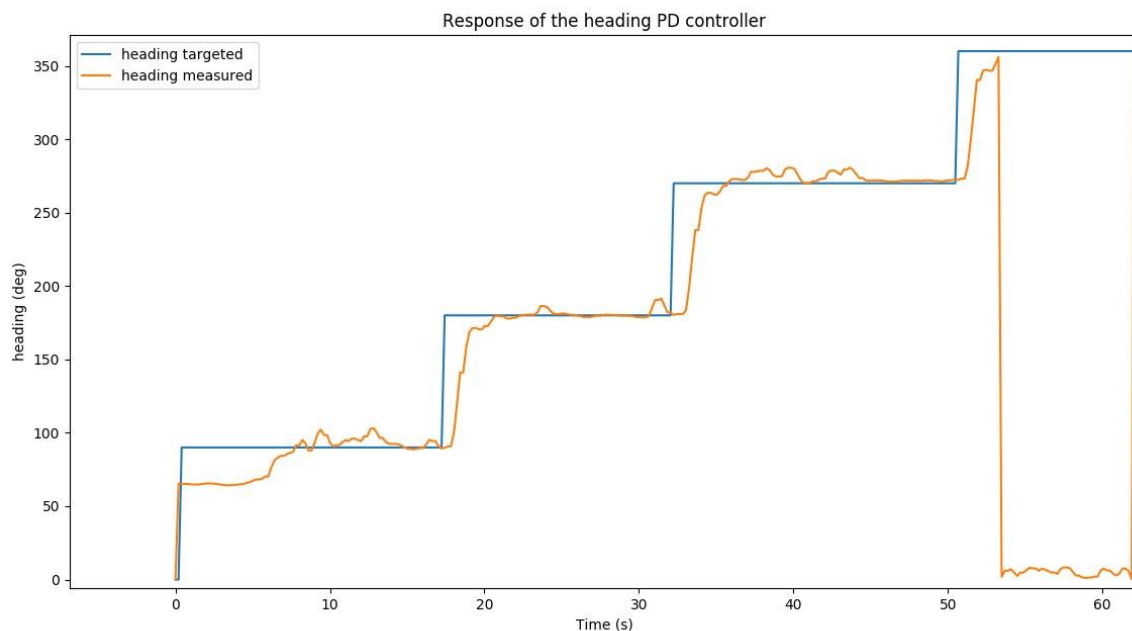


Figure 7 : Evolution of heading through time

On this example, the BlueRov2 did a square, so four 90° command in a row. The response time is quite low around 1s and it converges fast to the desired heading while being stable.

ii. Depth

We got the depth from the pressure measurement of the Bar30 sensor. As the pressure is linked to the depth in hydrostatic equation.

$$depth = \frac{-(p - p_0)}{\rho * g}$$

depth : current depth in meters

p : pressure measured in Pa

p0 : surface pressure (99 000 Pa)

ρ : water density (1000 kg/m³)

g : gravitational acceleration (9.81 kg/m²)

For the depth, we chose a PID controller. The command value is also filtered by the saturation function.

$$u = KI * I_depth + KP * (depth_desired - depth) - KD * D_depth$$

where I_depth is the integral of depth and D_depth its derivative.

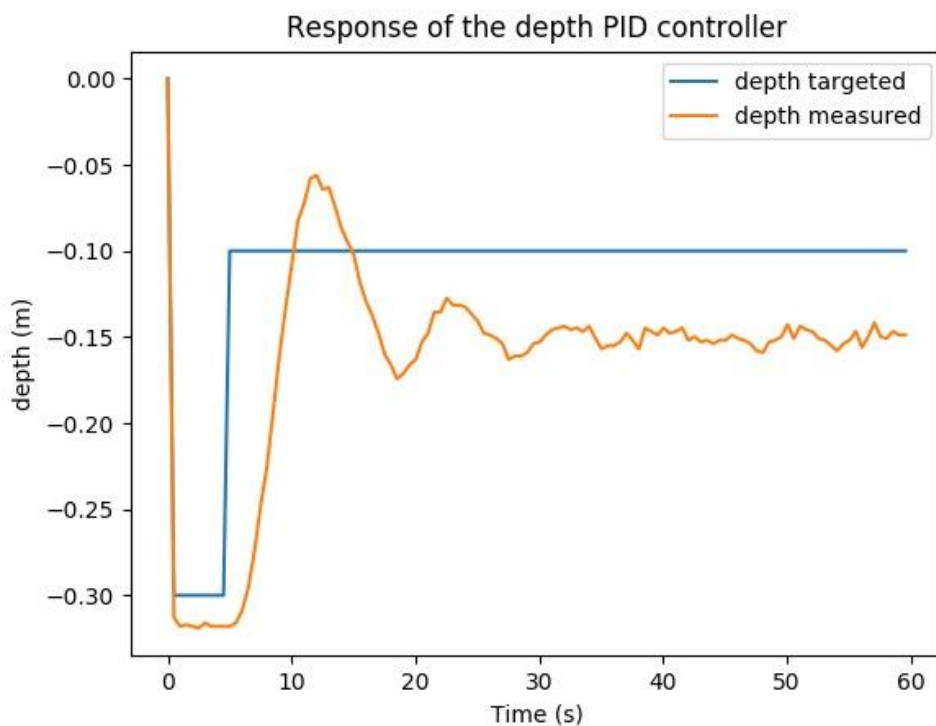


Figure 8 : Evolution of depth through time

Here the BlueRov2 was 30cm deep underwater, we asked to go at 10cm. The response is fast, and the depth is stable around 15cm which is 5cm lower than expected. The reason for that may be the weight of the robot. Indeed, as it comes closer to the desired depth, the command value decreases to the point where the motors are not running anymore and its weight makes it sink. When it sinks, it goes away from the desired position, so the command value increases. This pattern repeats itself.

To avoid that, we could have added a constant in the code that compensates this effect.

iii. Velocity

The velocity node is implemented in the ROS structure, but it only publishes a constant PWM. We originally wanted to implement a PD controller as we have acceleration data from IMU. However, the velocity estimate from integration is not a viable option. It is explained why in the following section.

3. IMU and velocity estimation

i. About IMUs

IMU stands for Inertial measurement unit, they are common sensors for robotics applications and of course used on the BlueROV2.

The BlueROV2 has 2 kinds of IMU:

- 1 PixHawk IMU for heading estimation
- 2 Adafruit LSM9DS1 IMUs for velocity estimation

The IMUs used have 9 DOF: 3 angular velocities, 3 linear accelerations, 1 magnetometer.

ii. Why 2 additional IMUs

We used 2 IMUs in order to take the mean value, and therefore gain in accuracy. It is also possible to use least squares approximation. Also, the sample rate of the Adafruit ones is much faster than the PixHawk.

iii. Raw IMU data computation

Data from relatively cheap IMUs are very noisy. The predominant error and noise sources are:

- constant bias error: average output of the device over a specified time, there are 2 parts one deterministic part called bias offset and a random part. The bias offset can be determined by calibration. The random part is a stochastic process and refers to the rate at which the error in an inertial sensor accumulates with time.
- bias instability: random variation in the bias computed over a finite sample of time
- angle random walk (gyros): white noise
- velocity random walk (accelerometer): white noise

To improve the quality of raw data we implemented an exponential filter (low-pass filter to cut white noise) and we did the calibration of the IMUs to remove the bias offset. However, we could not get rid of the random bias, which causes a drift after integration.

A solution to solve the issue would be to fuse IMU acceleration data with another measurement of velocity from another sensor (DVL, GPS...) in a Kalman filter. The IMU give a good short time estimation that needs to be corrected after a while with low frequency sensors, as GPS.

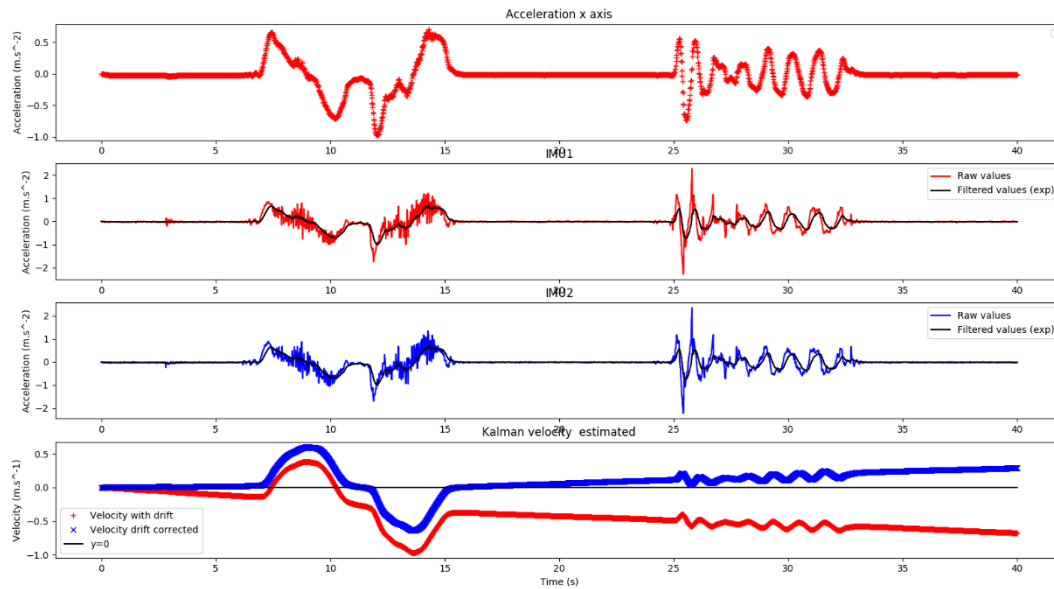


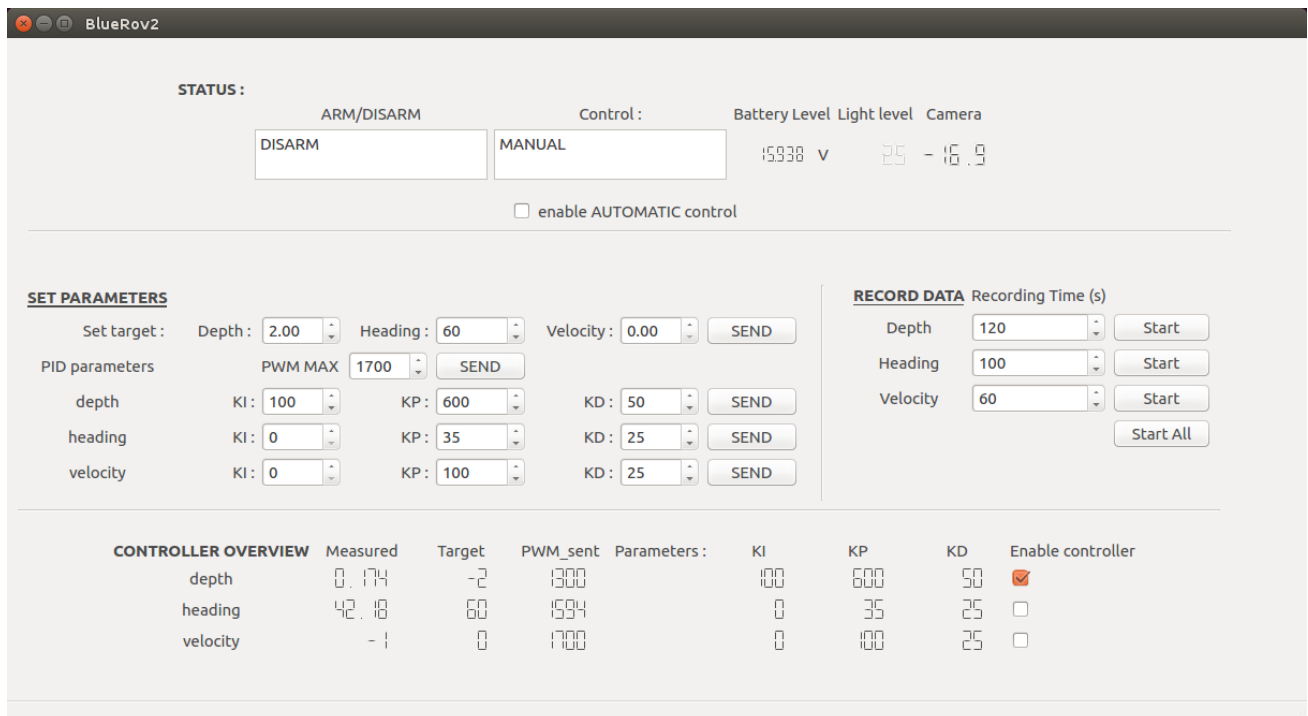
Figure 9 : Acceleration and velocity on x-axis

Those graphs were made based on data from the robot on a rolling table, we pushed the table forward, waited 10s and then backward to the initial position. The filter is quite efficient at removing noises however a drift is still observable after 30s. Therefore, it is not viable to estimate the velocity from acceleration integration only.

4. Graphical User Interface (GUI)

The GUI as many features:

- Check the status of the robot in real time.
- Set the controllers parameters easily and without touching the code.
- Record data from ROS topic using rosbag.
- A controller overviews.



The screenshot displays the BlueRov2 GUI interface. At the top, the 'STATUS' section shows 'ARM/DISARM' set to 'DISARM' and 'Control' set to 'MANUAL'. It also displays 'Battery Level' at 15.938 V, 'Light level' at 25, and 'Camera' at 10.9. Below this is a checkbox for 'enable AUTOMATIC control' which is currently unchecked.

The 'SET PARAMETERS' section is divided into two parts. On the left, 'Set target' includes Depth (2.00), Heading (60), and Velocity (0.00), each with a 'SEND' button. Below this, 'PID parameters' are listed for depth, heading, and velocity, with fields for KI, KP, and KD, and a 'SEND' button for each. On the right, 'RECORD DATA' includes 'Recording Time (s)' and three 'Start' buttons for Depth (120), Heading (100), and Velocity (60), along with a 'Start All' button.

The 'CONTROLLER OVERVIEW' section is a table with the following data:

| | Measured | Target | PWM_sent | Parameters : | KI | KP | KD | Enable controller |
|----------|----------|--------|----------|--------------|-----|-----|----|-------------------------------------|
| depth | 0.174 | -2 | 1300 | | 100 | 600 | 50 | <input checked="" type="checkbox"/> |
| heading | 42.18 | 60 | 1594 | | 0 | 35 | 25 | <input type="checkbox"/> |
| velocity | -1 | 0 | 1700 | | 0 | 100 | 25 | <input type="checkbox"/> |

Figure 10 : Graphical aspect

i. Status

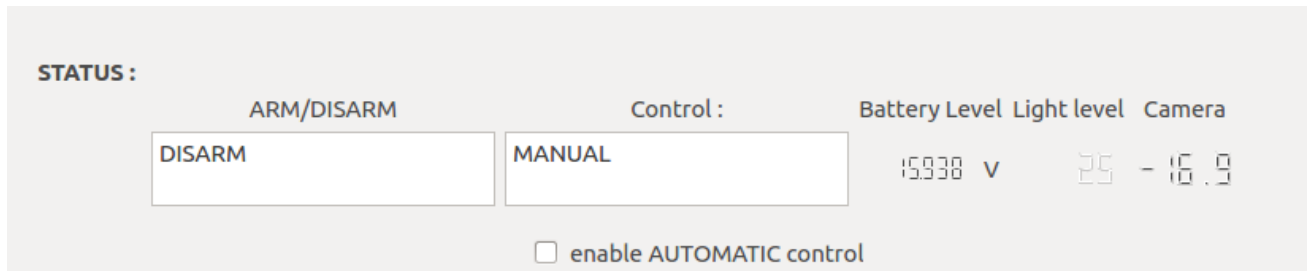


Figure 11 : Status

ARM/DISARM:

Show the arm status of the BlueRov2

Control:

- MANUAL: ROV controlled by gamepad
- AUTOMATIC: ROV controlled by the depth/heading/velocity controllers

enable AUTOMATIC control:

Allow to enable AUTOMATIC mode without a gamepad

Battery Level:

Display battery level in volt

Light level:

light level of the lights in %, 0: turned off, 100%: highest brightness

Camera:

Camera angle: between -45 to 45 deg

ii. Set parameters

SET PARAMETERS

Set target : Depth : 2.00 Heading : 60 Velocity : 0.00 SEND

PID parameters PWM MAX 1700 SEND

depth KI : 100 KP : 600 KD : 50 SEND

heading KI : 0 KP : 35 KD : 25 SEND

velocity KI : 0 KP : 100 KD : 25 SEND

Figure 12 : Set parameters

Set target:

- Depth: in m, 1m = 1m depth
- Heading: in deg, 0 is North
- Velocity: in m/s (the controller is not working)
- SEND: publish on /Settings/set_target

PWM MAX:

Set the PWM maximal for all controllers. Uses in saturation method in controllers.

PID parameters:

To tune PID for each controller.

KI, KP, KD = integral, proportional, derivative coefficients

iii. Record data

RECORD DATA Recording Time (s)

Depth 120 Start

Heading 100 Start

Velocity 60 Start

Start All

Figure 13 : Record data

This area allows to record data by launching a rosbag record command.

iv. Controller overview

| CONTROLLER OVERVIEW | Measured | Target | PWM_sent | Parameters : | KI | KP | KD | Enable controller |
|----------------------------|----------|--------|----------|--------------|-----|-----|----|-------------------------------------|
| depth | 0.174 | -2 | 1300 | | 100 | 600 | 50 | <input checked="" type="checkbox"/> |
| heading | 42.18 | 60 | 1594 | | 0 | 35 | 25 | <input type="checkbox"/> |
| velocity | -1 | 0 | 1700 | | 0 | 100 | 25 | <input type="checkbox"/> |

Figure 14 : Controller overview

It's an overview of the controllers with a preview of the controller's output in the column PWM_sent. The checkbox can be checked to enable the controller to drive the ROV if the AUTOMATIC mode otherwise it will not work until the AUTOMATIC mode is on.

5. Simulation and tests

While we implemented new functionalities, we had to test them to make sure it was working properly. We had two ways of doing that, by simulation and by several real tests.

i. Simulation

We used SITL, simulation in the loop. It shows the robot on a 2D map and include a physical model of water (drag forces, current ...)

We also had a 3D model with Gazebo, however it wasn't working properly so it wasn't that useful.

ii. Tests

For real tests, we had a tank of water (1.5 cubic meters). It was enough to check the heading and the depth controllers.

We did tests in real conditions, so in lakes. The mission to test our controllers was to do a square at constant depth.



Figure 15 : Set up for lake test and BlueRov2 in water

We also did tests for the IMUs, the robot was steadily placed on a rolling table and by moving the table we were able to observe accelerations evolutions on the x-axis (forward).

Conclusion

My internship at TUT has been a great experience. The finish culture is very different from the western Europe one. It's a calm country full of nature, lakes and saunas. People are kind when you get the chance to know them, indeed finish people don't talk much as smalltalk is perceived as rude. Working as engineer assistant meant working in autonomy with the support of the team. My robotics and tongue skills improved a lot. I also liked the research aspect of the internship, facing a technical problem and try to find a low-cost solution to was challenging and exciting.

List of Figures

| | |
|--|----|
| Figure 1 : Aerial view of the university | 7 |
| Figure 2 : aColor subsystems | 8 |
| Figure 3 : 2D Drawings..... | 10 |
| Figure 4 : ROS package structure..... | 12 |
| Figure 5 : Final ROS-graph | 13 |
| Figure 6 : ROS-graph early stage | 14 |
| Figure 7 : Evolution of heading through time | 15 |
| Figure 8 : Evolution of depth through time | 16 |
| Figure 9 : Acceleration and velocity on x-axis | 18 |
| Figure 10 : Graphical aspect | 19 |
| Figure 11 : Status | 20 |
| Figure 12 : Set parameters | 21 |
| Figure 13 : Record data | 21 |
| Figure 14 : Controller overview | 22 |
| Figure 15 : Set up for lake test and BlueRov2 in water | 23 |

Bibliography

Alex Becker (www.kalmanfilter.net (2016). *Online Kalman Filter Tutorial*. [online] Kalmanfilter.net. Available at: <https://www.kalmanfilter.net/default.aspx> [Accessed 28 Aug. 2019].

B. Rhudy, M. (2012). *Fusion of GPS and Redundant IMU Data for Attitude Estimation*. [online] Available at: https://www.researchgate.net/publication/258848233_Fusion_of_GPS_and_Redundant_IMU_Data_for_Attitude_Estimation [Accessed Aug. 2019].

Bancroft, J.B. and Lachapelle, G. (2011). Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors*, [online] 11(7), pp.6771–6798. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231672/> [Accessed 28 Aug. 2019].

Bluerobotics (2014). *Blue Robotics - ROV and Marine Robotics Systems and Components*. [online] Blue Robotics. Available at: <https://bluerobotics.com/> [Accessed 15 Aug. 2019].

Bluerobotics (2019). *ArduSub*. [online] ArduSub.com. Available at: <http://www.ardusub.com/>.

Dronecode Project (2015). *QGC - QGroundControl - Drone Control*. [online] QGroundControl - Drone Control. Available at: <http://qgroundcontrol.com/> [Accessed 15 Aug. 2019].

edmallon (2019). *Tutorial: How to Calibrate a Compass (and Accelerometer) with Arduino*. [online] Underwater Arduino Data Loggers. Available at: <https://thecavepearlproject.org/2015/05/22/calibrating-any-compass-or-accelerometer-for-arduino/> [Accessed 15 Aug. 2019].

ENSTA Bretagne (2018). *BlueROV2 Simple Autonomous Mission*. YouTube. Available at: <https://www.youtube.com/watch?v=jzzuCaQ6FTg> [Accessed 15 Aug. 2019].

Fabio Varesano (2012). *FreeIMU Magnetometer and Accelerometer Calibration GUI: Alpha Version Out*. [online] Varesano.net. Available at: <http://www.varesano.net/blog/fabio/freeimu-magnetometer-and-accelerometer-calibration-gui-alpha-version-out> [Accessed 15 Aug. 2019].

Gagne, D. (2019). *QGroundControl User Guide*. [online] Qgroundcontrol.com. Available at: <https://docs.qgroundcontrol.com/en/> [Accessed 15 Aug. 2019].

José Pereira, P. (2018). *bluerov_ros_playground*. [online] GitHub. Available at:

https://github.com/patrickelectric/bluerov_ros_playground [Accessed 15 Aug. 2019].

Kari T. Koskinen (2018). *Autonomous and Collaborative Offshore Robotics (aCOLOR)*. [online] Teknologiateollisuuden 100-vuotissäätiö. Available at: <https://techfinland100.fi/mita-rahoitamme/tutkimus/tulevaisuuden-tekijat/autonomous-and-collaborative-offshore-robotics-acolor/> [Accessed 15 Aug. 2019].

MAVLink (2009). *MAVLink Developer Guide*. [online] Mavlink.io. Available at: <https://mavlink.io/en/> [Accessed 15 Aug. 2019].

Nawaf, M., Merad, D., Royer, J.-P., Boï, J.-M., Saccone, M., Ben Ellefi, M. and Drap, P. (2018). Fast Visual Odometry for a Low-Cost Underwater Embedded Stereo System †. *Sensors*, [online] 18(7), p.2313. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6068653/> [Accessed 28 Aug. 2019].

Parrilla, M. (2019). *Inertial Navigation in Marine Robotics*. [online] Matthewparrilla.com. Available at: <https://www.matthewparrilla.com/post/inertial-navigation-marine-robotics/> [Accessed 28 Aug. 2019].

Salmony, P. (2019). *IMU Attitude Estimation – Philip Salmony*. [online] Philsal.co.uk. Available at: <http://philsal.co.uk/projects/imu-attitude-estimation> [Accessed 28 Aug. 2019].

tbabb (2015). *How a Kalman filter works, in pictures | Bzarg*. [online] Bzarg.com. Available at: <http://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/> [Accessed Jul. 2019].

Tedaldi, D., Pretto, A. and Menegatti, E. (2014). *A Robust and Easy to Implement Method for IMU Calibration without External Equipments - IEEE Conference Publication*. [online] Ieee.org. Available at: <https://ieeexplore.ieee.org/abstract/document/6907297> [Accessed 15 Aug. 2019].

The Qt Company (2019). *Qt Designer Manual*. [online] Doc.qt.io. Available at: <https://doc.qt.io/qt-5/> [Accessed 15 Aug. 2019].

Villa Escusol, J. (2018). *Air, Water Surface and Underwater Inspections with Just One Autonomous Robotic System*. [online] Teknologiateollisuuden 100-vuotissäätiö. Available at: <https://techfinland100.fi/acolor-autonomous-air-water-surface-and-underwater-inspections/> [Accessed 15 Aug. 2019].

Villa Escusol, J. (2019). *aColor Achieves First Year Goals of the Project with the Unmanned Surface Vessel as a Main Topic*. [online] Teknologiateollisuuden 100-vuotissäätiö. Available at: <https://techfinland100.fi/acolor-achieves-first-year-goals-of-the-project/> [Accessed 15 Aug. 2019].



RAPPORT D'ÉVALUATION 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

Cyril Cotsaftis

II - EVALUATION / ASSESSMENT

Veillez 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 ? (A) B C D E F
Was the initial contract carried out to your satisfaction?

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

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

ESPRIT D'ÉQUIPE / 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 :

Please evaluate the technical skills of the intern:

(A) B C D E F

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? oui/yes

non/no

Fait à _____, le _____
In Tampere _____, on 26.08.2019



JUSSI AALTONEN

Signature Entreprise _____

Company stamp _____



Signature stagiaire _____

Intern's signature _____

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

Table D - Traineeship Certificate by the Receiving Organisation/Enterprise
Tableau D – Attestation de stage de l'organisme/l'entreprise d'accueil

Name of the trainee: Cyril Cotsaftis

Nom du stagiaire :

Name of the Receiving Organisation/Enterprise: Tampere University

Nom de l'organisme/l'entreprise d'accueil :

Sector of the Receiving Organisation/Enterprise: Mechatronics

Secteur d'activité de l'organisme/l'entreprise d'accueil :

Address of the Receiving Organisation/Enterprise [street, city, country, phone, e-mail address], website:

Korkeakoulunkatu 6, 33720 Tampere, Finland, 03 311511, tau@tuni.fi, https://www.tuni.fi/fi

Coordonnées de l'organisme/l'entreprise d'accueil [rue, ville, pays, téléphone, adresse mail], site internet :

Start date and end date of traineeship: from [day/month/year]05/06/19..... to [day/month/year]30/08/19.....

Dates de début et de fin de stage : du [jour/mois/année].....au [jour/mois/année].....

Traineeship title: aColor - BlueROV2 ROS

Intitulé du stage :

Detailed programme of the traineeship period including tasks carried out by the trainee:

Programme détaillé de la période de stage, y compris les tâches réalisées par le stagiaire :

Implementation of a ROS architecture on the BlueROV2

Knowledge, skills (intellectual and practical) and competences acquired (achieved Learning Outcomes):

Connaissances, aptitudes (intellectuelles et pratiques) et compétences acquises (résultats d'apprentissage obtenus) :

ROS, python, command laws

Evaluation of the trainee: *Great work during the whole traineeship, including self-learning skills and good team work.*

Evaluation du stagiaire :

Date: 26.08.2019

Date :

Name, signature of the Supervisor at the Receiving Organisation/Enterprise and STAMP of the Receiving Organisation/Enterprise

Nom et signature du superviseur de l'organisme/l'entreprise d'accueil et TAMPON de l'organisme d'accueil

Jussi & Joven

JUSSI & JOVEN

