

SECOND YEAR INTERNSHIP REPORT

**COLLISION AVOIDANCE FOR AN AUTONOMOUS
SAILING BOAT**

**ÅLAND UNIVERSITY OF APPLIED SCIENCE
ENSTA BRETAGNE**

October 17, 2016

Student

SIMON CHANU
ENSTA BRETAGNE

Supervisor

ANNA FRIEBE
ÅLAND UNIVERSITY OF APPLIED SCIENCE

Abstract

A report on a second year internship in Åland University of Applied Science on collision avoidance detection for an autonomous sailing boat. Along the presentation and explanation of the functioning of Åland University, this report covers the research of a collision avoidance system for an autonomous sailing robot along with other minor jobs inside the team. The research began with a state of the art of the available collision avoidance algorithms which led to create our own on MATLAB. Then this algorithm was implemented and improved on the C++ code of the sailing boat command system. The simulation validated the code but it has not been tested in real condition. Overall this internship achieved its goal and was very instructive on code architecture and project management.

Contents

1	Presentation and goals of the internship	1
1.1	Åland University of Applied Science	1
1.1.1	Overview	1
1.1.2	Organisation inside Åland UAS	2
1.2	The internship	3
1.2.1	Goals and key issues in the internship	3
1.2.2	Quick presentation of the system	4
2	Design of a collision avoidance system	5
2.1	Teamwork methods and overview	5
2.1.1	Scrum and JIRA	5
2.1.2	Git	6
2.1.3	Presentation of the control system	7
2.2	Matlab design	7
2.2.1	Choice of the collision avoidance system	7
2.2.1.1	Voronoi Diagram	7
2.2.1.2	Vector fields	7
2.2.1.3	Minimum of potential field	8
2.2.1.3.1	Mathematical formula of the pikes and holes	8
2.2.1.3.2	Objective potential field :	9
2.2.1.3.3	Obstacle potential field :	9
2.2.1.3.4	Boat potential field :	10
2.2.1.3.5	Sum of these fields :	10
2.2.1.3.6	Adding of the sailing zone and the wind :	11
2.2.2	Condition of avoidance	12
2.2.3	Tests of the system	12
2.2.3.1	Change in the Matlab program architecture	12
2.2.3.2	Simulation inside Matlab	12
2.2.3.3	LabVIEW tests	13
2.3	C++ design	13
2.3.1	Sensor informations	14
2.3.2	Intersection of polygons	15
2.3.3	C++ tools for debug	15
2.3.4	Results	15
2.3.4.1	C++ code on Åland	15
2.3.4.2	the WRSC	16
3	Analysis of the internship	17
3.1	Åland University of Applied Science	17
3.2	Input of the internship	17
3.2.1	Knowledge learned	17
3.2.1.1	Scrum	17
3.2.1.2	Git	18
3.2.1.3	C++ architecture	18
3.2.1.4	Autonomous research work	18
3.2.2	What we could have done better	18
3.2.2.1	Message based architecture	18
3.2.2.2	The hardware	19

List of Figures

1.1	Large location of the Åland islands	1
1.2	Close location of the Åland islands	2
1.3	Organisation of the services inside Åland UAS	2
1.4	WRSC 2016 in Viana Do Castelo	3
1.5	4-meters boat called Janet	4
2.1	Scrum method for software development	5
2.2	JIRA	6
2.3	Presentation of our use of Git	6
2.4	Voronoi Diagram	7
2.5	Vector Field	8
2.6	Objective potential field	9
2.7	Obstacles potential field	10
2.8	Boat potential field	10
2.9	Sum of the potential fields without wind and sailing zone	11
2.10	Sum of the potential fields with wind and sailing zone	11
2.11	Matlab code architecture	12
2.12	Simulation of obstacle avoidance inside Mariehamn bay	13
2.13	Message architecture system by Jordan Less'Arđ Springet	13
2.14	Diagram of the messages exchanged among the collision avoidance system	14
2.15	Diagram of the detection of one obstacle	14
2.16	Simulation on VIBes-Viewer	16
3.1	Student accomodations found by the university	17

Chapter 1

Presentation and goals of the internship

1.1 Åland University of Applied Science

1.1.1 Overview



Åland University of Applied Science (Åland UAS), or Högskolan på Åland in Swedish is the University located in the town of Mariehamn in the Åland islands.



Figure 1.1: Large location of the Åland islands

They offers degree programme in Swedish in business administration, navigation, hospitality management, engineering (marine, electrical and IT) and health and caring science for a total of 600 students.

¹ Åland University of Applied Sciences was founded in 1997 by the Åland Provincial Government as a co-operative network for vocational higher education with the right to award polytechnical diplomas.

¹History taken from the great Åland UAS English website

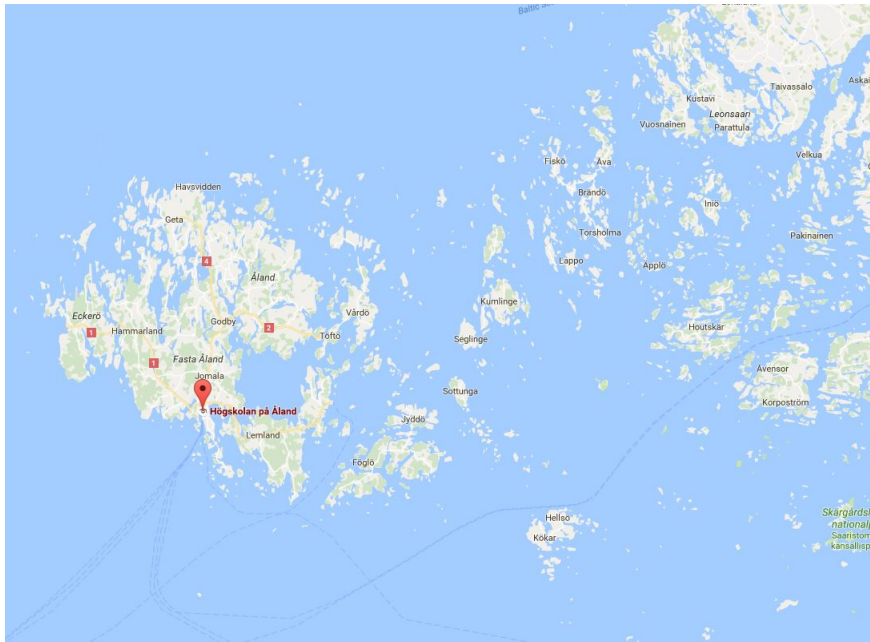


Figure 1.2: Close location of the Åland islands

On 1 January 2003, Åland University of Applied Sciences became a permanent institution when it merged with Åland Open University. After the merger, its Swedish name was changed to Högskolan på Åland. The English name has not been changed on the communication medium yet, that's why Åland UAS I will continue to call the university Åland UAS.

1.1.2 Organisation inside Åland UAS

The head of the university is Mr Edvard Johansson, as you can see on the diagram below, he oversees with his vice-director, Mr Ronny Eriksson all the departments of the University.

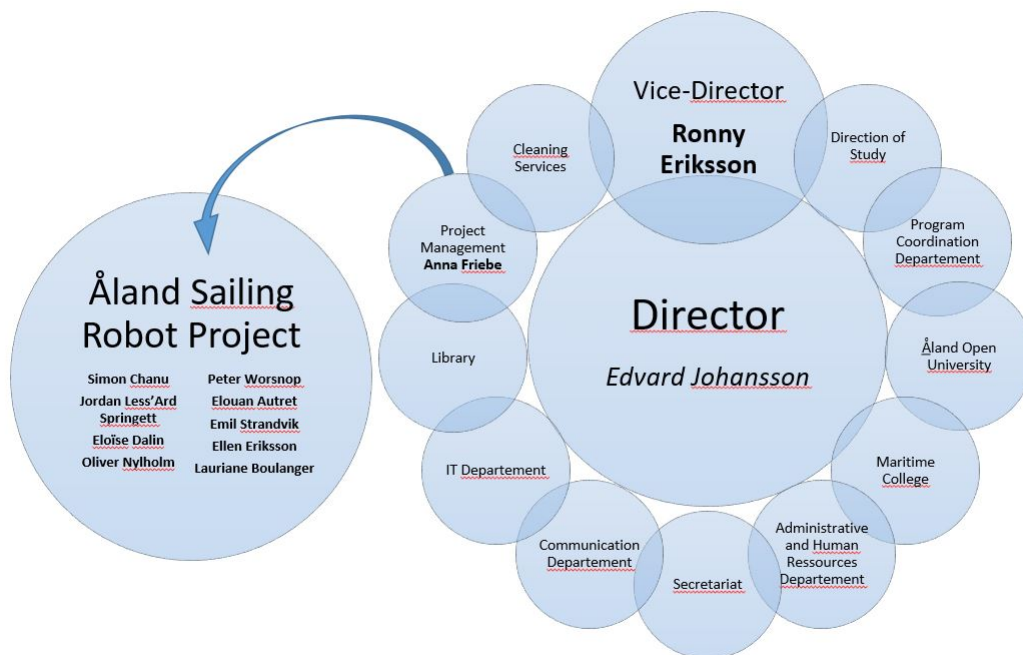


Figure 1.3: Organisation of the services inside Åland UAS

The Åland Sailing Robot Project, led by Anna Friebe as Project Manager at Åland UAS and overseen

by Ronny Eriksson as Vice-Director, is a project began in 2013 to create a cost efficient and environmentally friendly way to carry out autonomous task on the seas (marine research, surveillance, transport). This project is funded by the European Regional Development Fund and Åland UAS.

One of the final goal of the project is to create the first autonomous sailing robot to cross the Atlantic Ocean. Besides, each year Åland UAS participate to the World Robotic Sailing Championship (WRSC) with its software. This is a worldwide championship consisting of several challenges aiming to simulate the necessary features of a sailing robot.

1.2 The internship

1.2.1 Goals and key issues in the internship

My internship took place in Åland UAS in Mariehamn from the 20th of June to the 10th of September. At first my personal goal for this internship was to participate to the WRSC and fulfil the condition of going 12 week abroad during an internship. I applied in the Ålandish team because of several reasons : ENSTA Bretagne have good relations with them, they were first at the last WRSC, going to an archipelago was very appealing and last but not least, they were competing for the WRSC.

Initially, when we discussed the subject of the internship with Anna Friebe, my tutor, it was about the WRSC. I would prepare the boat for the competition by designing its collision avoidance system, and I would accompany the Ålandish team to the WRSC in Portugal to help them there.

Before the internship, the Åland Team and ENSTA Bretagne team merged into one team due to boat transport problem from Åland. ENSTA Bretagne would provide the boat, and Åland would provide the software. So it turned out that I wouldn't work against my school in the end.

The joint team decided to split the work in two. ENSTA Bretagne would provide the boat and the hardware, Åland UAS would provide the software.



Figure 1.4: WRSC 2016 in Viana Do Castello

As I explained in the presentation of the university, the WRSC is an international competition. It gives an international show of the quality of your work and contribute to the renown of a university. That's why even if the WRSC is not the main goal of the Åland Sailing Robot project some funds have been given to our team to participate to this competition. Moreover, Åland finished first last year, so we wanted to finish first this year as well.

1.2.2 Quick presentation of the system

I worked during my internship on a software designed for the boat below. This 4-meters boat is controlled by two actuators : one for the main sail and the other for the rudder. The electronic inside is composed mainly of a raspberry pi and the necessary hardware to get informations from the GPS, the wind sensors, the RC receiver and the 3G router. Working precisely the electronic architecture was not the purpose of this internship so the electronic architecture is in the annexes.

The software is composed of :

- A sailing logic : it regroup the high level commands of the boat. That's where I worked for most of the internship
- A waypoint manager : it's the system that retrieve the new waypoints from the database if the last ones were harvested.
- Drivers : this regroup everything that communicates with hardware (xbee, gps, wind sensor, actuators)
- An RC receiver : the boat had the ability to be controlled by a remote control. There is an RC multiplexer on the boat to switch from software control to manual control at any time.
- A database : it receive the logs from the system and contain parameters for the software.
- A website and an httpsync system on the boat : using the router to have an internet connection, the boat was synchronizing its database with the database of <http://sailingrobots.com/>. This website was used to monitor the boat and check the logs.
- A LabVIEW program communicating with the boat by xbee to monitor and take control of the boat. It can run a matlab script to command the boat.



Figure 1.5: 4-meters boat called Janet

This system changed completely for the WRSC but it will be explained later.

Chapter 2

Design of a collision avoidance system

2.1 Teamwork methods and overview

2.1.1 Scrum and JIRA

Our team was composed of 9 persons.

- Simon Chanu : specialized in robotics, worked on collision avoidance
- Jordan Less'Ard Springett : specialized in software engineering, rearchitected the whole code to a message based architecture.
- Eloïse Dalin : specialized in robotics, worked on the matlab part of collision avoidance and on the marine sensors.
- Oliver Nylhom : a video game programmer, worked on the C++ code and implemented the line following algorithm.
- Peter Worsnop, specialized in robotics, worked on the hardware communication and on the self steering wind vane.
- Emil Strandvik : specialized in web programming, worked on the web part of the project and was the scrum master of the team.



Figure 2.1: Scrum method for software development

We used for the software development the SCRUM method. It consist mainly in a backlog, which is basically the tasks left to do with different priority flags. Then the project is split in sprint of 2 to 4 weeks. AT the beginning of the sprint, the team choose together the tasks they should achieve during the sprint and plan approximatively the time needed for them. When the amount of time of the sprint is reached, the planning is done and the sprint can begin. During the sprint, the members can choose the task they want and help each other for a greater efficiency. Each morning we do a "daily scrum" : this is a quick meeting during which each member explains its current work and its planned work. At the end of the sprint, each member of the team present its work and the scrum master presents the results. Then a new sprint can begin from the backlog.



Figure 2.2: JIRA

We used JIRA to have a backlog and set up sprint. This is a very powerful tool online which allow each member to connect and change the status of a task, add comments or add new tasks in the backlog.

2.1.2 Git

The Åland Sailing Robot Project is an Open-Source research project. That's why it has a frequently updated github : <https://github.com/ÅlandSailingRobots/>

We used git in a formalized way explained by Jordan. The graphic below is an extract from one the internal documents on how to properly use git.

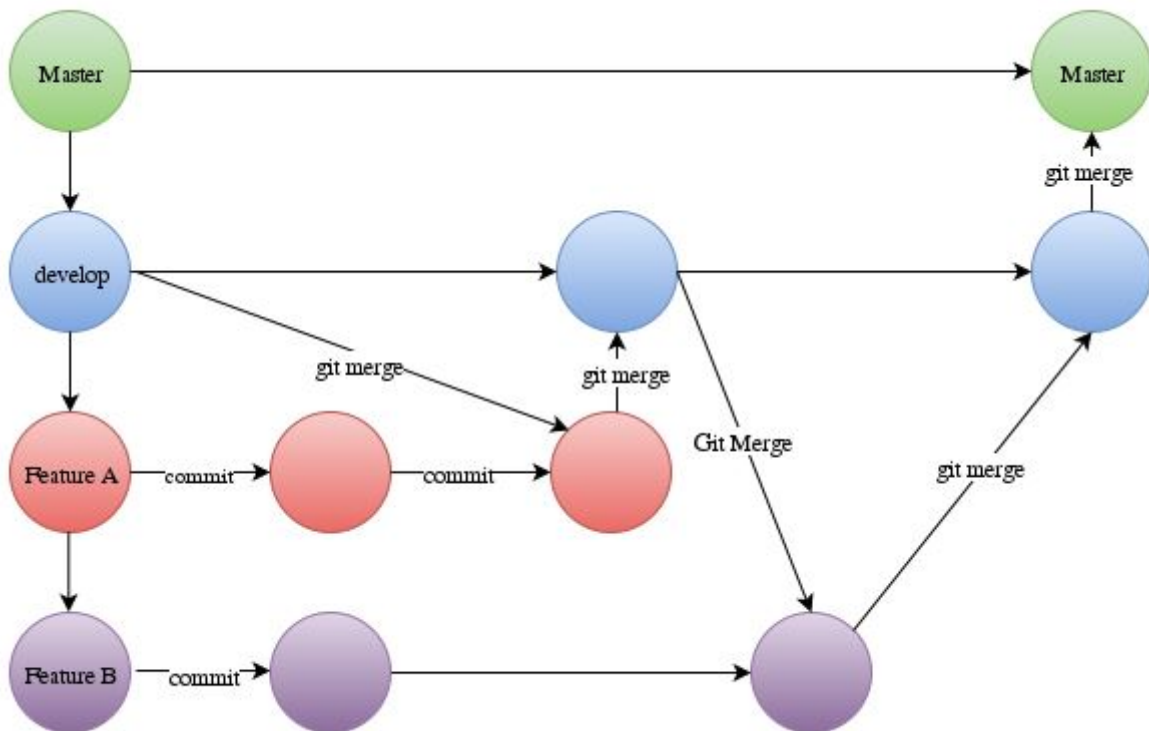


Figure 2.3: Presentation of our use of Git

The principle is to have a master branch always ready and working and other branch for specific development.

- The master branch contains only the code of develop that has been tested and validated in real conditions.
- The develop branch contains only code that compiles and has been through unit test and simulation if possible.
- The feature branch are the development branch. That's where the new features are coded and then merged into the rest of the code.

With this type of organisation, we figured out that we need to be careful not to have diverging branch : branch on which the code is so different that it become very difficult to merge them together.

2.1.3 Presentation of the control system

2.2 Matlab design

2.2.1 Choice of the collision avoidance system

I worked on the Matlab collision avoidance design with Eloïse Dalin from ENSTA Bretagne. We began by a state of the art of the different obstacle avoidance technics available, mainly from the work of Luc Jaulin. Among those which could fit to our problem, we can list Voronoï Diagrams and Vector fields command.

2.2.1.1 Voronoï Diagram

The voronoï diagram avoidance command is a simple and elegant method to avoid many obstacles at once. A voronoï diagram is created by finding the perpendicular bisector between each point to create tiles around the points. So the lines are the points equidistant from the 2 closest obstacles. In everyday life it can be seen on plants or bubbles.

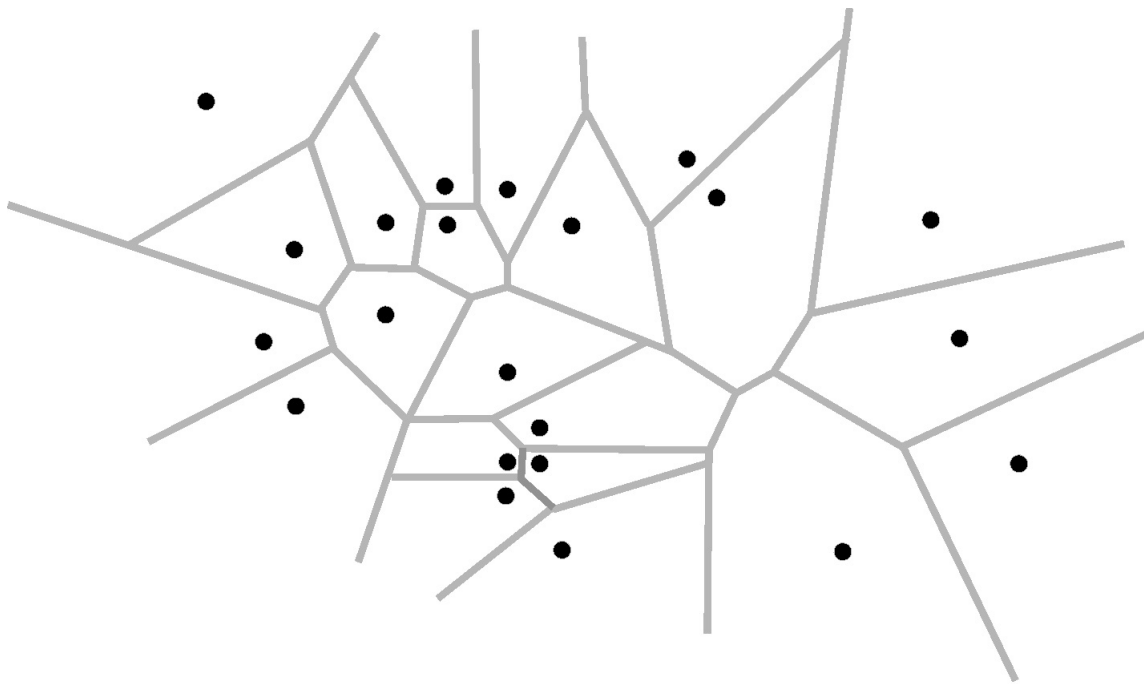


Figure 2.4: Voronoï Diagram

The command using voronoï diagram consist in following the lines created by the voronoï diagram, so the robot is always the furthest possible from the two closest obstacles.

The benefits of this system is that it's robust against many obstacles. Besides it allows us to use a line following algorithm for the boat which is what we want since following only waypoints doesn't correct the drift of the boat. On long travel such as crossing the Atlantic Ocean it could be a serious issue.

The drawbacks are that you need to have at least two obstacles to draw a voronoï diagram. In case of a lone obstacle this system wouldn't work. Moreover, if two obstacles are very close, there would still be a line between them that the robot could follow. But the last drawback is something that can be corrected by some conditions.

2.2.1.2 Vector fields

To get something that can handle one obstacles we looked into gradient based descent minimization of a potential field : following a vector field. To explain it simply, imagine that the robot is a drop of water, the obstacles

some mountains and the goal a wide basin. Then the drop will take the closest path avoiding the obstacles thanks to gravitation. This is the same idea here, the robot is following the least potential path.

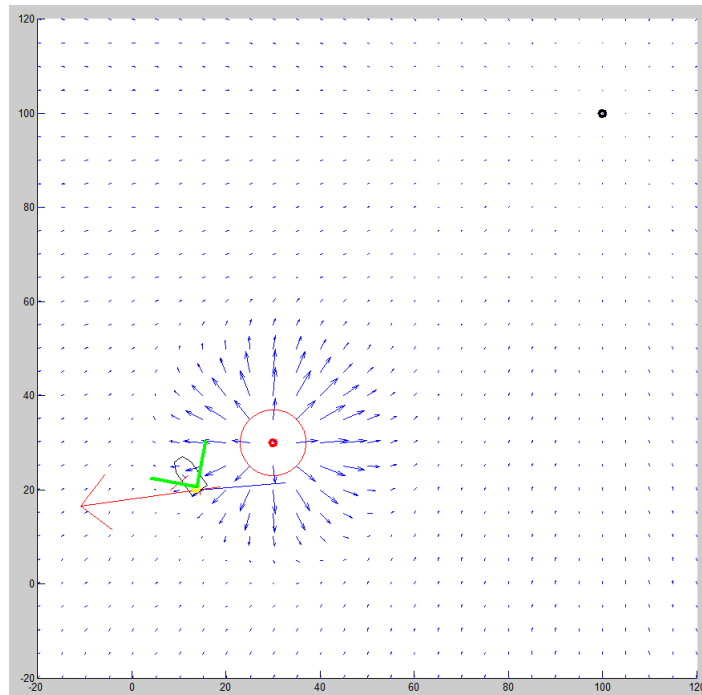


Figure 2.5: Vector Field

This is a very interesting method for ground robot because it can handle a lot of moving obstacles and is very flexible. It's possible to add custom potential to avoid some areas for example.

But in the case of a sailing robot, it can't just stop and go back when the boat is facing an obstacle. The boat will continue and crash into the obstacle, or turn heavily away from the closest path because of the wind. Moreover, the boat can't move forward facing the wind, so there is some headings it can face only briefly. That means in some cases the boat could be stuck in a position of constantly tacking from one side to the other. That's why the boat should follow a line long enough, more than three or four times the boat length, to be able to manoeuvre if necessary. That's not the case if it follows a vector field.

2.2.1.3 Minimum of potential field

According to the benefits and limitations of the technics we found, none was exactly fitting to our issue. So we decided to create our own system. The toughest system is the vector field command for its handling of lone obstacles. So we took this basis. The main problem was the use of vector in the command system. So we decided to keep the powerful and flexible tool that was the potential field and remove the gradient based descent minimization.

We decided instead to command the boat during the avoidance using a line following algorithm : the boat would follow a line and tack around it if necessary. A line is constituted by 2 points : a beginning and an end. The first point would be the boat at the collision detection, the second point would be the most interesting place to go in order to avoid the obstacles without going too far away from the ideal path. This last point would be computed by finding the minimum of the potential field.

The potential field is made of several summed potential fields. They have been created separately and then combined and calibrated to work properly together.

On the graphics below, the red color is the highest potential and the blue color the smallest.

2.2.1.3.1 Mathematical formula of the pikes and holes To modify a flat potential field, we decided to use the following function :

$$z = A * \exp(-t^2) \quad (2.1)$$

$$t = B * ((x + a)^2 + (y + b)^2) \quad (2.2)$$

With :

A the strength of the pike,

B the scale of the pike,

a the offset on the x axis and

b the offset on the y axis.

We used this function because it is simple and easy to use.

2.2.1.3.2 Objective potential field : The collision avoidance point should be the closest to the direction of the objective. So the potential field of the objective is wide and not very deep.

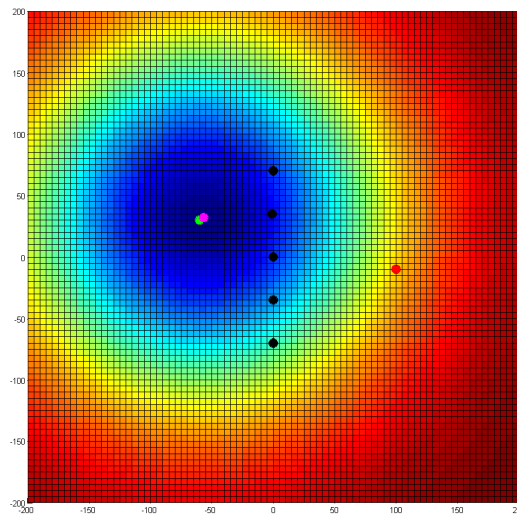


Figure 2.6: Objective potential field

2.2.1.3.3 Obstacle potential field : The collision avoidance point should be the furthest from any obstacles. Besides, the boat should avoid the obstacle by the left or the right at approximately 30 meters. So the obstacle is a big pike with one deep and narrow hole on each side. The obstacle is oriented in the direction of the line between the boat and the goal.

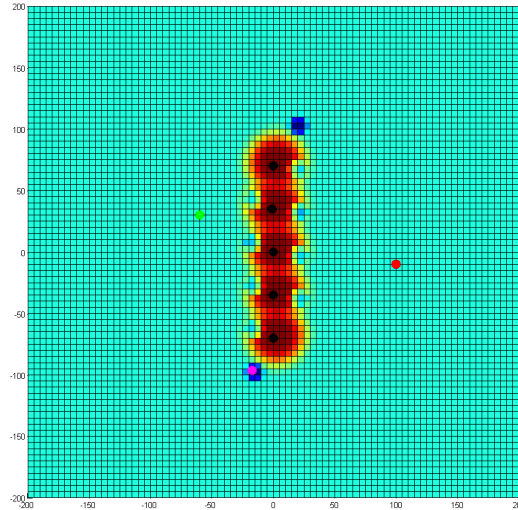


Figure 2.7: Obstacles potential field

After some testing we decided to cap the height of the obstacles to avoid very high values on the potential field. It was to make the debugging easier since the view is automatically scaled between the highest and lowest value.

2.2.1.3.4 Boat potential field : The collision avoidance point should be as close as possible to the boat up to a certain distance. Indeed, the collision avoidance point should make the boat avoid the closest obstacles, not the ones it has already avoided, nor the ones very close to the objective. However, in case of several obstacles drawing together a U shape with the boat at its center, the collision avoidance point is on the boat. That's not what we wanted because the boat wouldn't avoid any obstacles, so we added this pike on the boat.

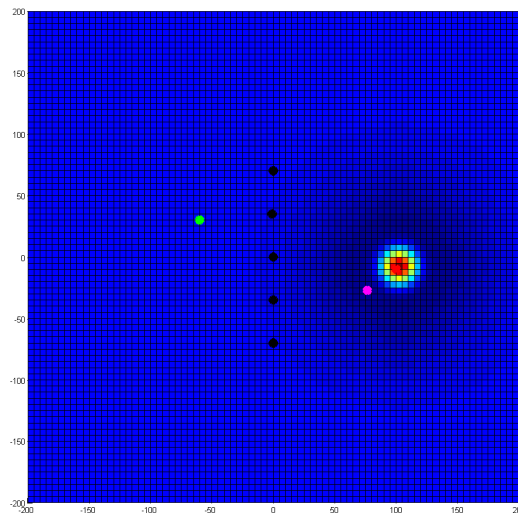


Figure 2.8: Boat potential field

2.2.1.3.5 Sum of these fields : After some calibration we obtained this result.

That's not all. The collision avoidance point must still stay on the water and be out of the No-go zone. The No-go zone is an area where the boat shouldn't go because its heading would be too close too the wind. Usually it's an infinite triangle of 90 degrees centred on the direction of the wind with the boat at its head.

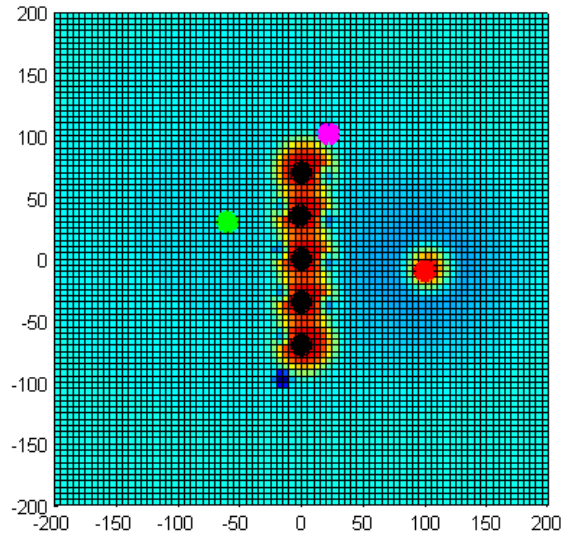


Figure 2.9: Sum of the potential fields without wind and sailing zone

2.2.1.3.6 Adding of the sailing zone and the wind : The No-go zone was easy to create, a simple product of step function of two different angles.

The same principle was used for the creation of the sailing zone. It's a product of step functions to create an area of potential equal to one. Then the axis is inverted to create a plateau of negative potential.

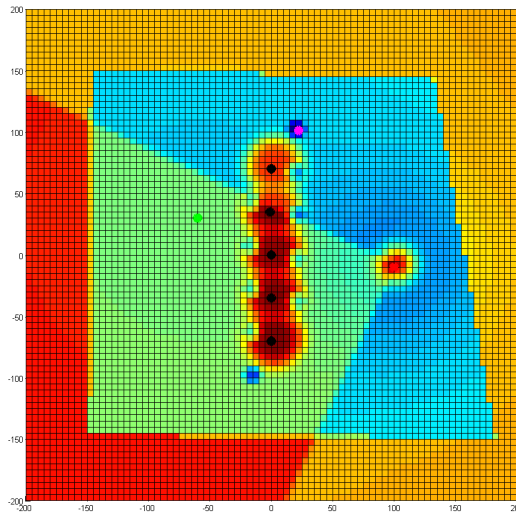


Figure 2.10: Sum of the potential fields with wind and sailing zone

The graphic above is the result of the calibration of all those fields. We can see the magenta point which point the minimum of the field, the red point which is the boat, the green point which is the objective and the black point which are the obstacles.

With simulations and tests we figured that the boat was not avoiding soon enough. So we decided to create the obstacles closer than their real position in order to simulate an anticipation from the boat. This is an offset of 5 meters.

2.2.2 Condition of avoidance

For a system to avoid an obstacle, a condition of avoidance is needed to prevent the boat from constantly avoiding anything or on the contrary, avoiding nothing.

At first we used the following condition :

If the boat is closer than 30 meters away from an obstacle and this obstacle has an intersection with the heading of the boat.

But in the command system of the boat we use a line following algorithm. that means the boat will follow the line or tack around it while staying below a predefined distance from the line. So if an obstacle was outside the channel created by this predefined distance but still visible by the boat while it's tacking. The boat will be likely to face the obstacle and avoid it even if there is no chance to hit it since it has no intersection to the channel.

So we decided to change the condition to this one :

If the boat is closer than 30 meters away from an obstacle and this obstacle has an intersection with the channel.

2.2.3 Tests of the system

2.2.3.1 Change in the Matlab program architecture

As explained before, to test this script in real condition we could use a LabVIEW program. Due to the size of our Matlab program we had to tidy the scripts in folders giving this result :

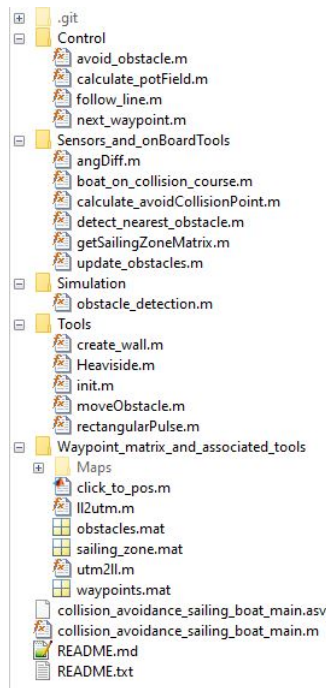


Figure 2.11: Matlab code architecture

With this architecture we have been able to keep the main for the LabVIEW and the main to test independently from LabVIEW separate.

2.2.3.2 Simulation inside Matlab

Once the system set up, we have been able to run some simulation. As explained in some text passages above, it has been useful to calibrate and change some parts of the algorithm.

In order to be able to use real gps points and interface with the LabVIEW program running on the boat we modified the Matlab script to use matrices of GPS points. These points were in utm coordinates. The utm

coordinates are the distance in meter from the origin point of an area on Earth.

So we have been able to simulate the behaviour of the boat with approximatively the same geographical parameters as the reality.

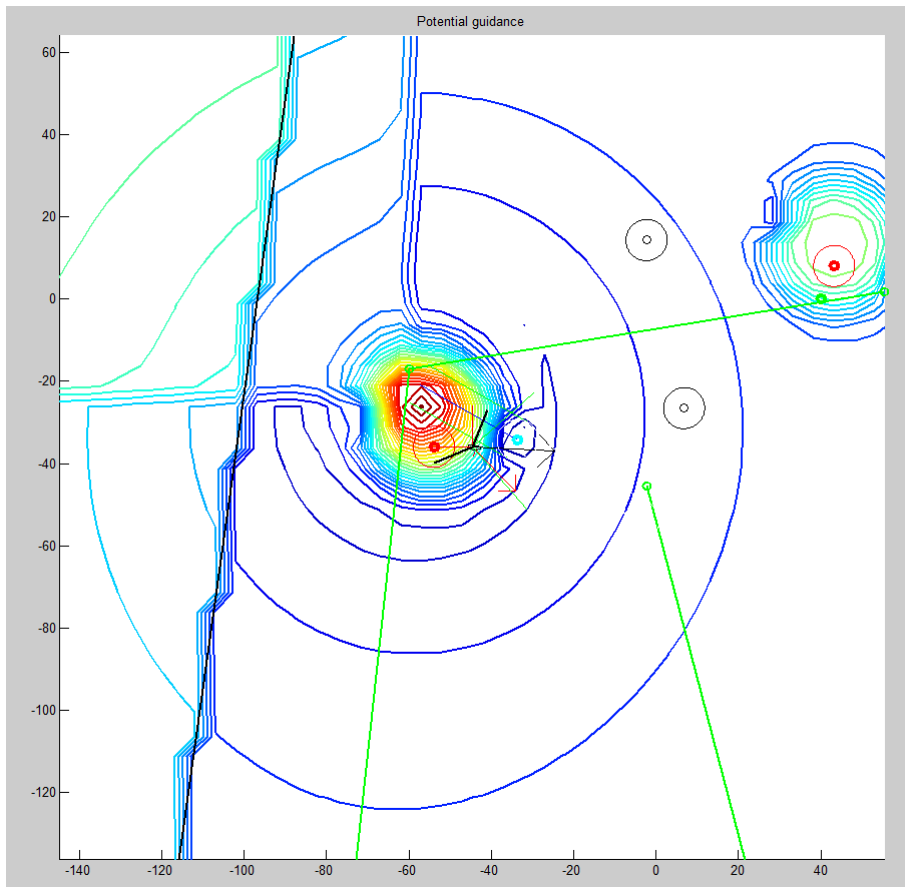


Figure 2.12: Simulation of obstacle avoidance inside Mariehamn bay

2.2.3.3 LabVIEW tests

If the Matlab simulation tests were conclusive, the tests with the LabVIEW program were not. None of us were really proficient in LabVIEW, this program had been inherited from students who worked on the project during the school year. I took some time to try to make it work, but by the end of the internship the WRSC was closing so we decided to give up the LabVIEW.

2.3 C++ design

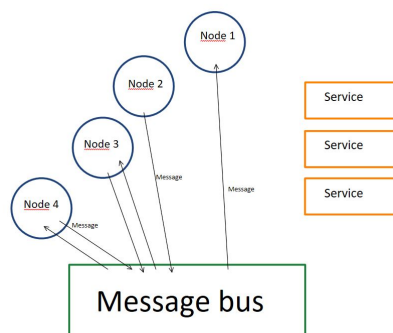


Figure 2.13: Message architecture system by Jordan Less'Arđ Springet

The C++ architecture of the boat have been heavily modified during my stay, we changed to a message based architecture programmed by Jordan Less'Arđ Springett. A message based architecture is composed of a message bus, which store and send the messages to the right part of the program, and of nodes, which are part of the programs processing an input (a message) and sending back an output (a message).

The collision avoidance system is programmed as a collision avoidance node. It receive the informations from the sensors, the waypoint manager and from the color detection node. It sends back waypoints to the waypoint manager node

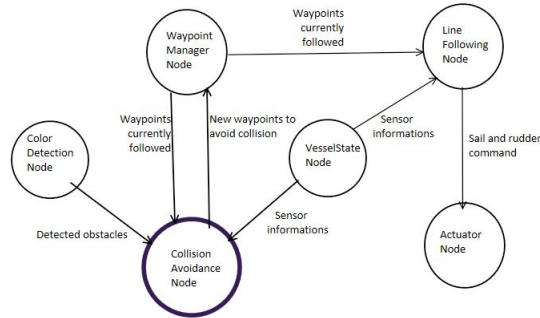


Figure 2.14: Diagram of the messages exchanged among the collision avoidance system

2.3.1 Sensor informations

When the Matlab design was approved I began to code the C++ version of the algorithm. It was approximatively at the same time that the team began to discuss the available sensors for the WRSC.

It turned out that we would use a boat too small to use any distance sensors. I would be retrained to a camera on the front of the boat. So I decided to use the same Matlab algorithm with a few additions and to program everything on C++ to spare some time.

I needed to find the distance to the obstacle with only the heading from the boat of the obstacle and the apparent width on the camera. So I decided to use the fact that the boat moves along time. The obstacles would be polygons that would be intersected with the new informations in order to make something similar to interval computation.

For example, on position A, the boat would see an obstacle so the system would memorize the area where the obstacle could be. Then, on a second loop, the boat would see the same obstacle from a different point of view and the system would then intersect the last triangle with the new one to get the closest center to the obstacle.

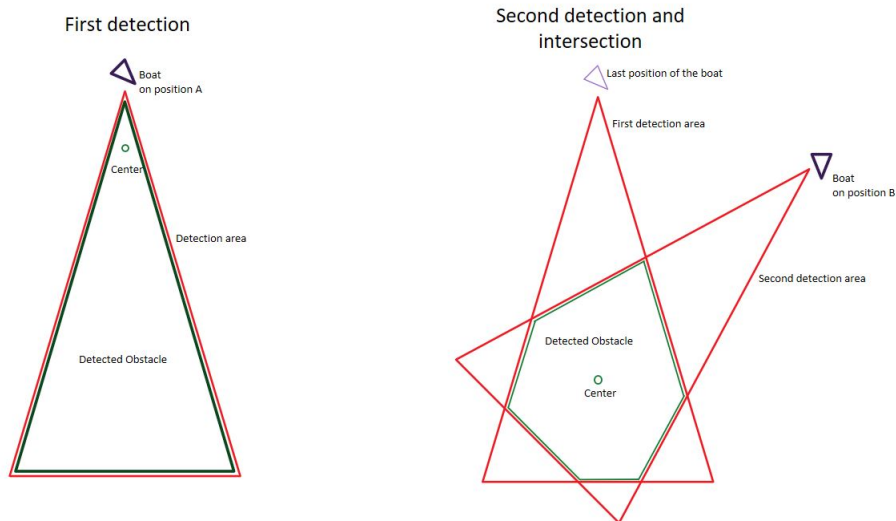


Figure 2.15: Diagram of the detection of one obstacle

In case of a first detection of an obstacle, that is to say a detection intersecting nothing, the center of the obstacle is computed as if the obstacle was a circle as close as possible inside the area.

2.3.2 Intersection of polygons

The polygons were stored as a class variable inside the code. To handle polygons I used the `boost::geometry` library since it's fast and very well documented.

The intersection algorithm work as the following :

- Fresh Data : the data from the obstacle detection are retrieved. It gives the heading of the obstacle and the angle measurement of its width.
- Cleaning : the obstacles that should have been seen and are not seen on the fresh data are erased from the memory
- Creation : the up-to-date obstacles are built from the informations of the fresh data.
- Merging : the up-to-date obstacles are merged with the memorized obstacles to refine the memory and get a more accurate position of the obstacle.

During the unit tests of my program, I figured out that boost doesn't handle very well the intersection of too much surfaces at the same time. If three surfaces overlap, then it crash. That's why I had to set up some safeties to prevent crashes. I had to split the merging part in three steps : first the up-to-date obstacles are fusionned together, then this union is intersected each by each with the memorized obstacles. Then, any up-to-date obstacle that hasn't been intersected is kept as a new one.

If anyone want to take a look at the code, everything is available on the github of Åland Sailing Robot on branch "CollisionAvoidance".

2.3.3 C++ tools for debug

To debug the program I used mainly the vibes drawing library developed by ENSTA Bretagne. This software for C++ allows the user to easily draw anything on another window from inside the program. With this software i have been able to show the behaviour of the intersection of polygons and correct bugs during the unit tests of each function.

Moreover, a C++ simulation of the boat, based on the model from Luc Jaulin has been created in python by Elouan Autret. It's a server that receive with over a socket the commands of the boat, and gives back to the simulation node inside the boat the new sensors informations. Elouan used a server/client architecture to be able to run the simulator on another computer while the system of the boat was running.

But this simulator didn't simulate the obstacles, so I had to modify the simulator to implement the obstacle simulation. Due to the limited amount of time we had before the WRSC I decided to code the simulation inside the simulation node instead of inside the simulator. It was because coding the transmission of the obstacle informations over the socket would have been too long.

So with the simulator and vibes, we have been able to simulate the C++ code on the boat. For example you can see below a screenshot of the simulation.

2.3.4 Results

2.3.4.1 C++ code on Åland

The programming of this part took a lot of time, around the half of my internship, but was eventually finished before the WRSC.

Unfortunately, there was still a bug inside the collision avoidance system. You can see on the screenshot of the simulation above that the red triangles are not where are the black dots. I spent the last week looking for this bug while programming a simple backup solution (If the boat sees an obstacle the line turns left).

I have since discussed with a hydrography teacher on this problem. Our boat was designed to cross long distances, so the command algorithms use a gps coordinate system which require a lot of computations to handle points and lines. For the collision avoidance system, I tried to lighten these computations. Since the obstacles aren't detected further than 100 meters from the boat, I used the gps coordinates as if it was meters using

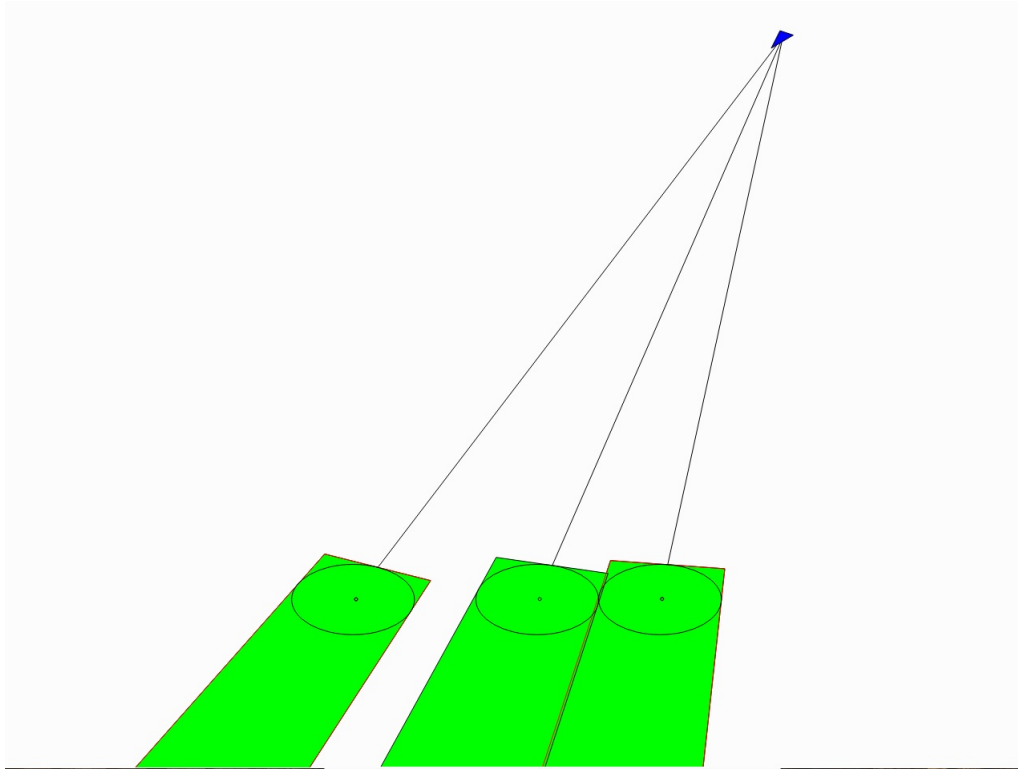


Figure 2.16: Simulation on VIBes-Viewer

a simple coefficient when it was necessary to transform latitude and longitude into meters. I measured this coefficient on a map at the equator, but I didn't figure out that this coefficient would change for the longitude according to the latitude of the gps point. This was the mistake. I have since sent an email to my tutor so she will be able to correct this mistake.

So before the WRSC I had a non-useable collision avoidance system and a backup collision avoidance system, besides a functional control system tested on a 4 meters boat.

2.3.4.2 the WRSC

Before the WRSC, the team met at Brest to share our work. There has been some issues on the hardware side of the project. The boat the joint team should have used wasn't available anymore. Moreover there was another boat available but it was impossible to use it if we weren't completely sure that the command system wouldn't break the boat. Indeed this boat doesn't belong to ENSTA either so some members of the team were very careful on that. Besides our control system had been designed for a 4 meters boat, and this boat was 1.5 meters long with a different hardware. We had coded the necessary driver nodes but nothing had been tested yet. So the team had to fall back on a much smaller boat.

During the competition, we used as well this smaller boat. It created a lot of issues we didn't see on the big boat. Indeed the control system of the 4 meters boat was slow enough to hide the remaining bugs inside the control system. This control system has not been coded by someone understanding the maths behind the command algorithm, so we had to recheck everything at the last moment. At the same time our modifications, probably due to stress and exhaustion, created more bugs. Fortunately, we managed to clean everything in the end.

But the size of the boat was still a problem. The WRSC took place in a river crossing Viana Do Castelo in Portugal. This river had a current of 1 meter per second and the wind was strong, creating big waves for this boat. So it was difficult, even on manual control to sail properly with this boat. So the control system never managed to sail properly, especially because the boat hardly tacked and gybed. Nonetheless, we used a monitoring tool all along the competition to verify the behaviour of the boat and the results were correct during the last two challenges.

Chapter 3

Analysis of the internship

3.1 Åland University of Applied Science

Åland UAS is a university funded by the Åland government and the subscription of its students to provide courses and fund research project such as the one I worked into. When I worked here I observed that Universities in France and in Åland function more or less the same way.

The particularity of this university is its size : around 600 students work here. This is tiny for an average university. It entails that every people knows each other so there is a small hierarchical distance between us and the different members of the administration. Moreover, the internship took place during the summer, so there was very few young people on the University. So even if some of them were responsible for handling my accommodation and some other administrative tasks, they were close with the team.



Figure 3.1: Student accomodations found by the university

The most surprising part was before the internship. The administration of the university handled everything for me. They lent me a bike, took care of the accommodation and the transport from the airport. During the internship they accepted late payment for the rent of the flat and were worried about the European grant I should have received sooner in the internship.

3.2 Input of the internship

3.2.1 Knowledge learned

3.2.1.1 Scrum

On this internship I really learned how to work efficiently in a small team. The scrum method is certainly something I will remember and use in the future.

With this method we have been able to track our work and the work of others very easily. Besides, the use of daily short meeting allowed us to keep track of works far from our area. For example, when one of the member of the team working on a self-steering wind vane for the 4-meters boat took a leave of 1 week, I switched from my work to his because it was urgent. I have been able to go into his work very quickly because I knew

approximatively what he did.

Moreover, whenever someone had some issues on a task, he would tell it in the daily scrum meeting and someone would give him some advice.

Overall, it's a very effective way of working on software for a small team on small projects. However, the team needs an effective task manager software such as JIRA to use the Scrum method at it's full potential.

3.2.1.2 Git

During the internship, I our C++ specialist, Jordan Less'Ard Springett taught the team on how to properly use git. I already explained the way of using it in 2.1 TEAMWORK METHODS AND OVERVIEW.

It gave us the tools to use git at it's full potential. This is a valuable knowledge I will use in my projects in the future.

I also learned a practical example of why a README file and specifications documents are so important. Because when you join an ongoing project, getting to know how to work on the code is very difficult without these files. For example, at the beginning of the internship it took me 2 days to install the code on my computer because I had found no installation document. So I had to ask to other members of the team and try to handle the problems myself. So a README file on github is mandatory on a team project.

3.2.1.3 C++ architecture

In Åland I learned a new programming language : C++. I discovered a fast and efficient language with lots on available libraries on the internet to help on particular tasks. For instance : polygon handling or display. However this is an old language with a lot of specificities that make it difficult to master. For example, an object is always copied when as an argument of a function, that's why most function uses only the references of objects. This specificity makes the code much harder to read.

For this project, even if python would have greatly reduced the development time, I don't think it would have been suitable. Indeed, the collision avoidance system handled 100x100 matrices for the potential field and handled polygon intersections. In python, the computational time would have been under 0.1 seconds. However, for quick and little projects I won't use C++ in the future since the set up time to create something is much longer.

3.2.1.4 Autonomous research work

The management in Åland was very free, we had a lot of autonomy : it was very instructive. Anna Friebe didn't verified our work closely to make sure everything was right trusting us to be able to speak if something wasn't right. So we had to be liable and autonomous in a foreign language.

3.2.2 What we could have done better

3.2.2.1 Message based architecture

We decided to change from the old architecture to a custom built message based architecture to improve the maintainability of the code while keeping a light architecture. That's why Jordan designed a message based architecture based on a queue storing pointers to the nodes.

However, this architecture has been built from scratch so very few tools have been developed to monitor and simulate easily and quickly the system. That's something very important that I learned during the internship : **SPLM**

- **Simulate**

Being able to easily and quickly simulate a system is a major necessity. For example in Åland, when we tested our code on the 4-meters boat it took one day to set up everything and make the test. So testing several times a day some slight changes in the algorithm is out of question.

- **Playback**

That's something that lacked the system during the WRSC. On several challenges, some behaviour of

the sailing robot were strange. We noticed it on the monitoring system but we have never been able to precisely explain them. A playback system associated with a logging system would have solved that.

- **Log**

During the WRSC the system had a logging system. But showing them took at least 20 minutes. We should have set up an easier log display to avoid losing time.

- **Monitor**

The monitoring system is totally mandatory on an autonomous robot. Indeed, we need to know if the behaviour of the robot is logical or not. Our system worked with a configurable monitoring tool developed for the Abersailbot team

There is some existing middleware that uses these tools. ROS or MOOS for example. We decided to create our own message based architecture to avoid such heavy middlewares. But in the end we didn't have the necessary and easy to use tools to debug effectively the system. I would advise for the future to heavily improve this software with these tools or to switch to a known middleware such as MOOS or ROS.

3.2.2.2 The hardware

There was some little boat of the size of the one we used during the WRSC at the university. We tried to create an electronic for this boat to be able to test our algorithms on a much smaller robot.

Apart from the lack of electronic tools on the university that slowed a lot the creation of this hardware, it has never been finished. Indeed the WRSC closing in we decided to completely finish the software instead of setting up a new boat. It was a mistake. Indeed the change of scale of the boat is very important : the wind and the waves don't scale down. Besides, there was some bugs we haven't been able to see on the 4-meters boat we saw on the small boat as explained in the results of the internship. So when we ended up with a small boat we haven't been able to change everything in time to make it work properly.

So I have learned that the tests must be made since the beginning on something close to what we will use in the end. Besides, the two teams should have worked more closely so we would have been able to test sooner when the Aland team arrived to Brest.

Conclusion

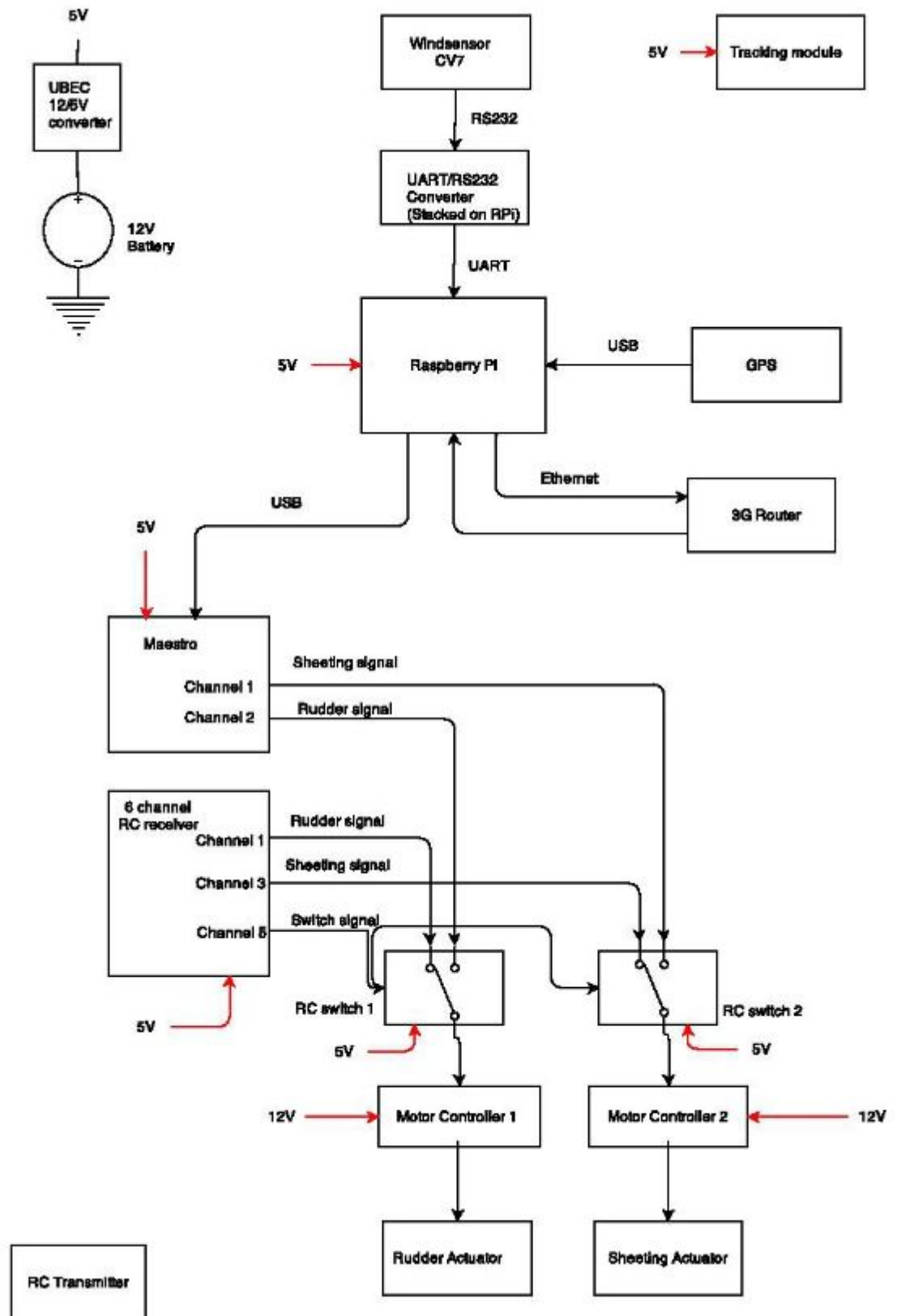
This traineeship in Aland University of Applied Science has been a very instructive experience. I worked on the exiting project of developing the artificial intelligence for collision avoidance for a sailing robot in a great environment. I began at first by creating the algorithms on Matlab, then while improving it, I translated everything to C++ on the boat system. So I have created a complex system from the beginning to the final tests on a sailing boat. There has been a lot of issues during the internship but I learned a lot of technical skills with git, C++, but management skills as well with Scrum. In the end, while doing a work that will be used in the future, this great internship has been very instructive on project management and robotic projects.

Bibliography

- [1] Åland University of Applied Sciences - In English - Högskolan på Åland. www.ha.ax.
- [2] Luc Jaulin. *La robotique mobile*, pages 0 – 274. ISTE Editions, 2015.
- [3] Luc Jaulin and Fabrice Le Bars. *A Simple Controller for Line Following of Sailboats*, pages 117–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [4] Chris Venes. Calculate distance, bearing and more between latitude/longitude points. <http://www.movable-type.co.uk/scripts/latlong.html>.

ANNEXES

Electronic Architecture



INTERNSHIP AGREEMENT

HOST ESTABLISHMENT NON-SUBJECT TO FRENCH LAW

Between:

ENSTA Bretagne - Ecole Nationale Supérieure de Techniques Avancées Bretagne,

2 rue François Verny

29806 BREST Cedex 9 France

Represented by Mr Patrick Puyhabilier, duly authorized in his capacity as Director

Hereinafter referred to as "ENSTA Bretagne"

And

HOST ESTABLISHMENT :

ALAND UNIVERSITY A.S.

Aland University of Applied Sciences

BP 1010, AX-22111 - Aland

MARIEHAMN FINLAND

Represented by Mr/Ms/Mrs/Miss Renny Eriksson duly authorized in his/her capacity as Vice rektor

Hereinafter referred to as the "Host Establishment",

And

Mr **Simon CHANU** student of ENSTA Bretagne (**Civilian student**)

Nationality : French

Born on 21 October 1994 at ST RENAN (29) - FRANCE

N° Sécurité Sociale (National Insurance Number) : 1942926023275 75

Title of Course : CI - Année 2 - Branche SPID -

Year: 2015 - 2016

Profile : Systèmes - perception - information - décision

Hereinafter referred to as "The Intern",

2 rue François Verny
29806 Brest Cedex 9
Tél. +33 (0)2 98 34 88 00
Fax +33 (0)2 98 34 88 46



École Nationale Supérieure de Techniques Avancées Bretagne - Membre du Groupe ENSTA



Hereinafter collectively referred to as "The Parties",

The following has been agreed :

ARTICLE 1 : SCOPE

The present agreement determines the conditions of organization and procedure of the internship accomplished by The Intern within the Host Establishment.

ARTICLE 2 : THE PEDAGOGICAL PROJECT AND INTERNSHIP CONTENTS

2.1 Expectations of the Internship in Terms of the Pedagogical Project, the Objectives and Results:

The internship is compulsory. It is part of the training program of ENSTA students in the first year/second year/third year/masters/post masters degree.

An internship program is established in common agreement between the two parties, according to the training received by The Intern at ENSTA Bretagne. The aim is to enable The Intern to experience practical training which will complement and complete the theoretical training.

Theme of the internship: (to be completed by the Host Establishment)

Atand sailing Robots

2.2 Activities Entrusted to The Intern (to be completed by the Host Establishment):

ARTICLE 3 : ORGANIZATION OF THE INTERNSHIP

The internship takes place from **20 June 2016** to **10 September 2016**. Under no circumstances, may the internship end after graduate.

The duration of the internship can be modified by simple amendment of the present agreement.

Upon request of ENSTA Bretagne, The Intern may occasionally interrupt his internship (viva voce etc). The Host Establishment will be forewarned of these dates.

ARTICLE 4 : SUPERVISION

The Host Establishment will appoint a training supervisor **Mrs Anna FRIEBE** in charge of technical supervision and optimization of the internship conditions of achievement. The pedagogical supervisor appointed by ENSTA Bretagne is :

- Ms Natacha CAOUREN.

ARTICLE 5 : DISCIPLINE AND ABSENCE

During the internship and his presence on the premises of the Host Establishment, The Intern is subject to the in-house rules of the Host Establishment, notably those concerning working hours, safety and medical visits.

In the case of non-respect of the in-house rules of the Host Establishment, on the part of The Intern, the internship may be terminated. Any disciplinary action falls within the jurisdiction of ENSTA Bretagne, the Host Establishment will provide the constituent elements should this be the case.

The Registrar of ENSTA Bretagne will be notified of any absence.

ARTICLE 6 : THE GRATUITY

6.1 Military Student

The military student may not receive a monthly gratuity.

6.2 Civilian Student

For the duration of the internship, the Host Establishment will make provision for the payment of a monthly gratuity of _____ euros/dollars/other (*to be completed by the Host Establishment*).

Travel and accommodation costs incurred by The Intern on behalf of the Host Establishment, as well as training costs that may be required by the internship, are wholly taken care of by the Host Establishment, according to the procedure in force in the Host Establishment.

List of proposed advantages :

- _____
- _____
- _____

ARTICLE 7 : SOCIAL WELFARE

7.1 Work-Related Accident

Should The Intern fall victim to a work-related accident due to or during the internship, the Host Establishment and/or The Intern will immediately inform ENSTA Bretagne and transmit the elements necessary for the declaration and processing of the work-related accident.

7.1.1 Military Intern

During the internship period, the military Intern will be considered to be on duty. He is thus covered by the French State for any accident that could arise during the internship and associated travel. ENSTA Bretagne will declare the work-related accident to the appropriate services, according to the information communicated.

7.1.2 Civilian Intern

The coverage of the risk of professional accidents and illness depends on the gratuity paid to The Intern.

i) Gratuity less than or equal to 13.75% of the French hourly earnings threshold (the internship being considered as non-remunerated as regards the French laws of social welfare) :

The coverage of work accidents and work-related illnesses of The Intern affiliated to the French regime is maintained for a maximum period of twelve (12) months, in accordance with Article R.444-7 of the French Code de Sécurité Sociale (Code of Social Welfare). The Intern continues to benefit from the specific work-related accident coverage in Article L. 412-8 (2°) of the French Code de Sécurité Sociale, student regime, relative to work-related accidents. In the case of an accident, the Director of the Host Establishment undertakes to alert ENSTA Bretagne (Registrar) as soon as possible, to indicate the circumstances of the accident so as to enable a declaration to be made to the relevant Caisse Primaire d'Assurance Maladie (National Health Insurance Office). These steps are identical for The Intern receiving no gratuity.

ii) Gratuity greater than 13.75% of the French hourly earnings threshold (the internship being considered as remunerated as regards the French laws of social welfare) :

To be advised by the Host Establishment (delete where inappropriate)

- *The host country has the benefit of a work accident and work-related illness protection scheme and the Host Establishment pays contributions pertaining to coverage of these risks. In the case of a work-related accident, The Intern will be able to benefit from social welfare protection in the host country.*
- *The host country does not have the benefit of a work-related accident and work-related illness protection scheme. The Intern is strongly advised to subscribe to a private insurance scheme.*

7.2 Responsibility for Medical Expenses

7.2.1 Internship carried out in an EU¹/EEA² country or Switzerland³

Before leaving to do the internship in an EU/EEA country or Switzerland, whatever the amount of the gratuity paid, The Intern must apply for a European Health Insurance Card (EHIC) :

- i) If the intern is a civilian, from his Caisse Primaire d'Assurance Maladie (National Health Insurance Office) at least two (2) weeks before his departure ;
- ii) If he is a military Intern, from the Caisse Nationale Militaire de Sécurité Sociale (National Military Health Insurance Office), at least twenty (20) days before his departure.

This card is valid for a maximum of a year. It attests Health Insurance rights during the stay. The EHIC (or if urgent, the Provisional Replacement Certificate – PRC, valid 3 months) guarantees the payment of medical care necessary during a stay abroad, under the same conditions as those of insured citizens in the host country.

7.2.2 Internship outside an EU/EEA country and Switzerland

i) Military Student

Medical expenses

Medical expenses incurred by The Intern, military student, in a foreign country outside the EU/EEA and Switzerland, are covered by the Caisse Nationale Militaire de Sécurité Sociale (National Military Health Insurance Office). Thus in application of Articles R.761-12 and R.761-13 of the Code de la Sécurité Sociale (Code of Social Welfare), the military intern benefits from the provision, in kind, of health insurance and maternity coverage throughout his carrying out of duties abroad, under the terms and according to the refund rates made provision for in the general welfare regime for adherents on assignment abroad.

Medical Transport and Repatriation

Only transport expenses incurred in the host country are refundable. This notwithstanding, medical repatriation costs are covered in the case of serious accident or acute phase of an illness occurring unexpectedly, which cannot be treated in situ and which require hospitalization in France.

iii) Civilian Student

Medical Expenses

Certain countries have an agreement with France⁴. The Intern should contact his health insurance organization in order to obtain information as to the terms of social welfare protection.

For an internship outside an EU/EEA country/Switzerland, in a country which has not signed an agreement with France, only urgent medical expenses **may** be covered by the Caisse d'Assurance Maladie (National Health Service), or mutuelle étudiante (mutual student insurer) on the presentation of bills or receipts to the limit of the refund rates in force in France.

It is in The Intern's best interest to ensure his complete social welfare protection

PCI2ASPID1 15 / Simon CHANU

- by subscribing to private insurance from his “mutuelle etudiante” (mutual student insurer)
- or adhering to a student welfare protection scheme in the host country
- or by taking out insurance from the Caisse des Français à l’Etranger (www.cfe.fr) (French Expatriates’ Health Office)
- Medical Transport and Repatriation

The Intern is strongly advised to take out international health and repatriation insurance in case of illness abroad.

ARTICLE 8 : PUBLIC LIABILITY AND INSURANCE

Each of the three Parties declares that they are covered for Public Liability.

Injuries or accidents experienced by The Intern and attributable to the Host Establishment or its personnel, are compensated for according to the rules in force in the Host Establishment and/or the place of internship. The details of these rules are available to The Intern on simple request.

The Host Establishment undertakes to pay for any damage which could be caused by The Intern in the course of his internship duties. The Host Establishment accepts that it will not initiate proceedings against The Intern, or ENSTA Bretagne, to seek compensation for this damage, except in the case of willful error.

Should the Host Establishment make a vehicle available to The Intern, it is responsible for ascertaining beforehand that the insurance policy covers the vehicle’s use by The Intern.

In the case of personal liability, without a direct link to the service or the internship, The Intern can be called upon to bring his personal public liability insurance into play.

ARTICLE 9 : CONFIDENTIALITY

The Intern is subject to the rules of confidentiality and/or loyalty in force in the Host Establishment.

ARTICLE 10 : INTERNSHIP REPORT, VIVA VOCE AND EVALUATION

The internship will be the subject of a report and a viva voce. The mark attributed will be integrated into the global marking scheme.

For students following a second year or third year pedagogical curriculum, wherever possible, a representative of the Host Establishment will be a member of the viva voce jury.

The Intern will also forward a copy of his report to his internship supervisor. The latter will inspect the contents, notably in respect of Article 9, “Confidentiality” and forward his evaluation to ENSTA Bretagne who will take this into account in the attribution of a viva voce mark.

At the end of the internship, the Host Establishment will deliver a certificate of achievement to The Intern, indicating the nature and duration of the internship.

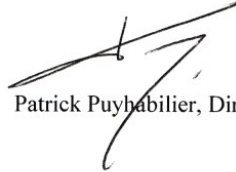
Made in Brest, in three copies, one for each Party

Date: 21/04/16

The Intern,



For ENSTA Bretagne,



Patrick Puyhabillier, Director



For the Host Establishment,

Aland UAS

Date: 4.5.2016

Representative of the Host Establishment

(name and position)



Host Establishment Stamp



¹ **EU : European Union (28)** : Austria, Belgium, Bulgaria, Cyprus (southern zone under the control of the Government of the Republic of Cyprus, excepting the northern zone under the control of the Turkish Army and excepting the buffer zone under the control of the UN) the Czech Republic, Croatia, Denmark, Estonia, Finland, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, The Netherlands, Poland, Portugal (including the archipelagoes of the Azores and Madera), Romania, Slovakia, Slovenia, Spain (the Iberian Peninsula, the Balearic Islands and the Canaries), Sweden, the United Kingdom (England, Scotland, Wales, Northern Ireland and Gibraltar).

² **EEA : European Economic Area** = EU +Norway + Iceland + Liechtenstein (These countries are not part of the European Union but have concluded a cooperation agreement on the European Economic Area).

³ **The Swiss Confederation** concluded an agreement with the European Union on the free movement of persons, which entered into force on 1 June 2002.

⁴ **List of countries having concluded an agreement with the French Sécurité Sociale (National Health Service)**: Algeria, Andorra, Benin, Bosnia-Herzegovina, Cameroun, Canada, Cap-Verde, Chili, Congo, Côte d'Ivoire, the United States, Gabon, Israel, Jersey, Macedonia, Madagascar, Morocco, Mauritania, Monaco, Mali, Niger, Nouvelle Calédonie, the Philippines, Polynesia, Québec, San Marin, Senegal, Togo, Tunisia, Turkey, the Federal Republic of Yugoslavia.



RAPPORT D'EVALUATION ASSESSMENT REPORT

Merci de retourner ce rapport dès la fin du stage à / Please return this report at the end of the internship to :

ENSTA Bretagne – Bureau des stages - 2 rue François Verny - 29806 BREST cedex 9 – FRANCE

☎ 00.33 (0) 2.98.34.87.70 - Fax 00.33 (0) 2.98.34.87.90 - stages@ensta-bretagne.fr

I - ORGANISME / HOST ORGANISATION

NOM / Name *Åland University of Applied Sciences* _____

Adresse / Address *PB1010, AX-22111 Mariehamn, Åland, Finland* _____

Tél / Phone (with country and area code) *+358 (0)18 537000* _____

Fax / Fax (with country and area code) *+358 (0)18 16913* _____

Nom du superviseur / Name of the person in charge of the placement

Anna Friebe _____

Fonction *Project manager, Åland Sailing Robots* _____

Adresse e-mail / E-mail address *anna.friebe@ha.ax* _____

Nom du stagiaire accueilli / Name of the trainee

Simon Chanu

II - EVALUATION / ASSESSMENT

Veillez attribuer une cote, en encerclant le chiffre approprié, pour chacune des caractéristiques suivantes. Cette note devra se situer entre **A (très bien)** et **F (très faible)**
Please give a mark between A (very good) and F (very weak).

MISSION / TASK

❖ La mission de départ a-t-elle été remplie ? A B C D E F
Has the task been carried out well ?

❖ Le stagiaire a-t-il apporté des connaissances nouvelles à l'organisme d'accueil ? oui/yes non/no
Did the trainee bring news skills to the host organisation ?

Lesquelles ? *Which ones ?* _____

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

Si oui, lesquelles ? *Which ones ?* _____

ESPRIT D'EQUIPE / TEAM SPIRIT

❖ Le stagiaire s'est-il bien intégré dans l'organisme d'accueil / Did the trainee easily integrate into the host organisation?
(disponible, sérieux, adaptation au travail de groupe)
(flexible, dedicated, adapts himself (herself) to the team work) A B C D E F

Avez vous des observations ou suggestions à nous faire part / *Do you have any remarks or suggestions to comment ?* *Simon was dedicated and a valued member of the international team.*

COMPORTEMENT AU TRAVAIL / BEHAVIOUR TOWARDS WORKS

Le comportement du stagiaire était-il conforme à vos attentes (*Ponctuel, ordonné, respectueux, soucieux de participer et d'acquérir de nouvelles connaissances*) ?

Did the trainee come up to expectations (*Punctual, methodical, responsive to management instructions, concerned with quality, concerned about gaining new skills*) ?

(A) B C D E F

Avez vous des observations ou suggestions à nous faire part / Have you any remarks or suggestions to comment ?

INITIATIVE – AUTONOMIE / INITIATIVE – AUTONOMY

Le stagiaire s'adaptait vite à de nouvelles situations ?

(*Proposer des solutions aux problèmes rencontrés, autonome dans son travail,..*)

(A) B C D E F

Did the trainee adapt himself (herself) well to new situations ?

(*Suggest solutions to problems met, was independent in his/her job...*)

(A) B C D E F

Avez vous des observations ou suggestions à nous faire part / Have you any remarks or suggestions to comment ?

CULTUREL – COMMUNICATION / CULTURAL – COMMUNICATION

Le stagiaire était-il ouvert, d'une manière générale, à la communication

(*Was the trainee open to listening and expressing himself (herself)*)

(A) B C D E F

Avez vous des observations ou suggestions à nous faire part / Have you any remarks or suggestions to comment ?

OPINION GLOBALE / OVERALL ASSESSMENT

❖ La valeur technique du stagiaire était :

(*The technical skills of the trainee were :*)

(A) B C D E F

III - PARTENARIAT FUTUR / FUTURE PARTNERSHIP

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

(*Are you prepared to host another trainee next year ?*) oui/yes

non/no

Fait à _____, le _____ In
Mariehamn _____, on October 11, 2016 _____

Signature

Uma Feiele

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

