Development of an Autonomous WAM-V for the Maritime RobotX Challenge 2014

Team WaZa – Tokyo Institute of Technology

Andre Yuji Yasutomi, Takaaki Kitamori, Alexis Holgado, Arturo Ceron-Lopez, Jianhua Li, Jose Antonio Silva Rico, Jose Luis Torres Arroyo, Atsushi Horigome, Shunichi Kurumaya, Akihiro Ohno, Guan-Horng Liu, Satoshi Kitano, Edwardo F. Fukushima

1. Abstract

This paper summarizes all the work the team WaZa has done to develop an autonomous WAM-V and participate in the Maritime RobotX Challenge 2014. It starts explaining all the hardware developments mentioning the mechanical, electrical and electronic developments, and finishes explaining the software approach mentioning the software modularization, the task strategies and the navigation control algorithm used. In the mechanical developments, it shows why we chose to use the four outboard motors for propulsion instead of just two, and shows the motor coupling designed and the idea of installing the batteries in the skies. In the electric developments it shows the electric box designed for this project as well as the safety system developed to ensure the safety of the users. In the electronic developments, it shows the sensor used for the Vision System, the Pinger Detection System and the positioning and heading system which are all enclosed in what we call the electronic box. In the software approach, it shows how we divided the software in modules and made all the integration using the RT-Middleware. It shows also the logic we used to tackle all the tasks of the competition, the sensor data filtering we did using the Kalman Filter and moving average, and the Position, Speed and Heading Control developed using Fuzzy Control. The paper finishes explaining the lessons we learned when developing for the first time a big scale robot and explaining some of the issues we had to deal with.

2. Introduction

WaZa is the team representing Tokyo Institute of Technology. Its name means “technique” in Japanese, but also is an abbreviation of Water Zamurai, a name that relates with Samurai, the Japanese warriors of feudal Japan.

The team consists of members with different specialties, educational background, nationalities (8 in total) and culture, forming a heterogeneous and experienced team.

The objective when participating in this competition was not only to obtain an experience in an international competition or to use the work in this competition as a research theme for a graduation projects, but to win the competition.

To do so, we used the strategy of transferring all our knowledge of robotics we obtained with our researches to the development of this autonomous boat. This helped us making the project faster once some of the developments needed for this project were already done in our research projects.

3. Technical Approach

In this session it will be explained all the work we have done to develop our autonomous WAM-V which is shown in Figure 1.

Figure 1 - WAM-V - Team WaZa

It will start explaining about all the hardware developments, divided into mechanical, electrical and electronic developments, and will finish with the software approach and the task strategies.

3.1. Mechanical Developments

Once the boat provided to the team for the competition did not come with any kind of propulsion system, the objective for the mechanical designs was the selection of the appropriate propulsion system, the design of a suitable coupling to attach the motors and the design of cases to protect some sensors and electric elements against the water.

3.1.1. Motor Choice

A comparison between different number and type of motors, including the Torqeedo Motor which is commonly used for this type of boat, was done in order to select the most suitable and advantageous configuration (Table 1).

Model	Torqeedo C.2.0 24V	CAYMAN B 12V	CAYMAN Pro 24V		PROTRUAR 2HP 24V	
# of motor	2	2	2	4	2	4
Total thrust [lbs]	345	110	160	304	200	380
Total weight [Kg]	36.6	36.4	36	72	17	34
Final Speed [m/s]	5.66	2.2	3	5	3.6	6.1
Time to reach 90% of Final Speed [s]	1.93	2.09	2.2	2.8	1.9	2.3

Table 1 - Motor Comparison

After analyzing different options, we decided to use two Protruar 2HP 24V motors per hull. Those motors, when together, showed a similar performance than the Torqeedo motors but with the advantage that two motors per hull provides the system with a redundancy which might be useful in case one of the motors stop working for any unfortunate reason. In this case, the boat could still keep running with the remaining motors having only to stop one of the motors in the other hull.

3.1.2. Motor Coupling

The motor coupling was design to fit with the hinge that was provided with the boat, the coupling is made of aluminum frames, with these frames we are able to do some modifications in case of any necessary adjustment. The motor couplings were equipped with floaters to keep the motor floating in a specific level. To increase the actuation of the floaters and reduce the length of the coupling we decided to place the floaters behind the motors.

Figure 2 - Coupling without motor (left), coupling with motor (right)

3.1.3. Battery Cases

Due to the batteries used to power all the system are Lead-acid and this kind of batteries are considerable heavy, we decide to place them on the skies in order to have easy access to them when they need to be exchanged. For this reason the case was made of stainless sheets to have a rigid and water proof box.

Figure 3- Battery Case

3.2. Electrical Developments

3.2.1. Electric Components Description

The electric distribution system of the boat uses four 12V G&Yu sealed lead-acid batteries, each with a capacity of 115Ah, to power it. Those batteries were divided into two power units, in which the batteries are connected in series to provide 24 volts to the boat. The two power units are connected then in parallel in order to have a capacity of 230 Ah which can provide power to the system during 1.5 hours in continuous operation.

The electric components, excluding the batteries, are contained in a waterproof Takachi box, hereon referred to as Electric Box (Figure 4). The Electric Box is fixed onto the Payload Tray of the Catamaran Boat. For connections outside the Electric Box, IP67 Takachi connectors are used.

Figure 4 - Electric Box (External view with Connectors and DC Breakers)

To turn on and off the whole system, two 150A rated DC Breakers are used as general switches. All the electric and electronic systems can be turned OFF with these DC Breakers.

To prevent the batteries from charging each other, due to small variation of voltage, a pair of diodes is incorporated into the system design (Figure 5).

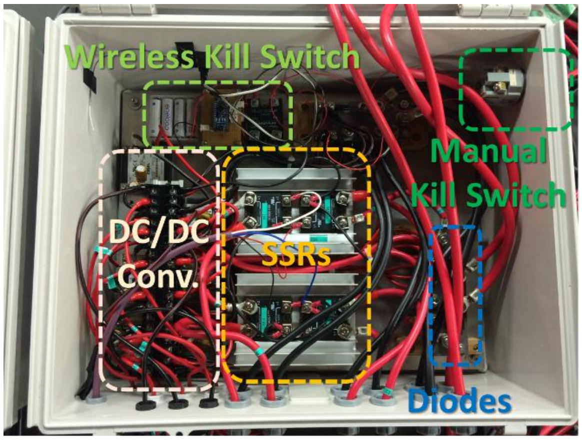

Figure 5 - Electric Box (Internal view)

For each of the four motors used to move the boat, a Solid State Relay (SSR) from Crydom is used. Each SSR controls the turning ON and OFF of its respective motor.

The Electric Box powers the Electronic Box. To prevent damage of the electronics, DC-DC converters are used to deliver power to the Electronic Box. Six DC-DC Converters were implemented, two of 5V, three of 12V and one of 24V. The first tests with the DC-DC Converters showed that because of the inductance of the long cables used, the voltage of the output had a variation with the expected voltage. To solve this, capacitors were used in the input and output of the DC-DC Converters which regulated the voltage.

3.2.2. Kill Switch

For safety reasons and in accordance with the competition rules, a Kill Switch System is used. The Kill Switch is contained in the Electric Box and has a Manual and Wireless deactivation mode (Figure 5). An Arduino Microcontroller, with independent power, controls the Kill Switch System. The Wireless activation mode uses a HobbyKing Transmitter and Receiver. When the Manual or Wireless Deactivation is made, the Microcontroller sends a signal through a SSR that turns OFF the power of the four motors. When the reset is made through Wireless or Manual mode, the Microcontroller activates back the motors.

3.3. Electronic Developments

3.3.1. Propulsion System

To control the propulsion system there was the necessity of connecting it to a microcontroller which would perform the low level control and communicate with the embedded PC. The motors chosen for the propulsion system didn’t provide with the angular velocity measurement, having thus the necessity to have an external circuit to measure it. There was also the necessity of measuring the current consumed by the motors and transmitting all the information via Ethernet to the embedded PC.

To fulfill all this necessities, the circuit shown in Figure 6 was designed and manufactured.

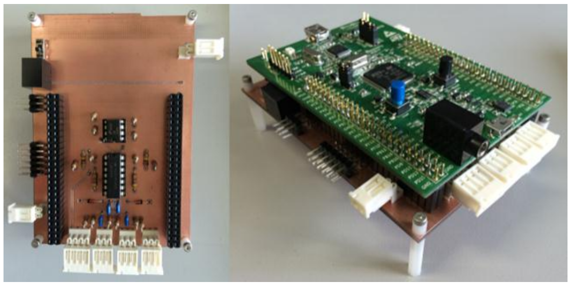

Figure 6 - Circuit for Propulsion System

This circuit once connected to a STM32f4 discovery microcontroller is capable of:

- ✓ measuring motor's angular velocity
- ✓ measuring motor's current
- ✓ Sending analog signals to control the motors
- ✓ Acquiring the signals from a remote control receiver
- ✓ Communicating with other devices via CAN and Ethernet

As it was decided to use 4 motors for the propulsion system and this circuit can control up to 2 motors, it was necessary to use two of those circuits with their respective microcontrollers.

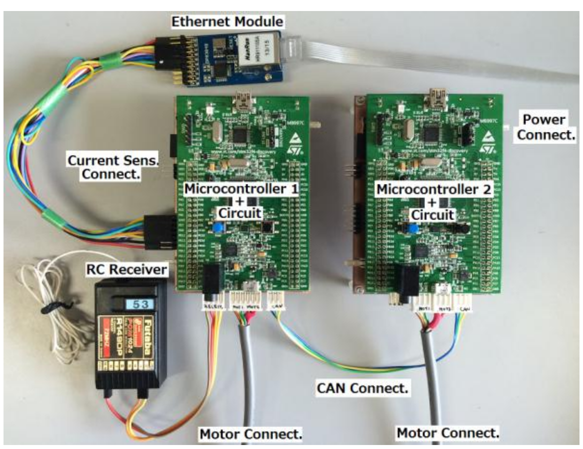

Figure 7 - Propulsion System Electronic Configuration

The communication between them was performed by CAN and the communication with the embedded PC was performed by Ethernet with the help of an external DP83848 Phy breakout board. The whole electronic system for the Propulsion System is shown in **Error! Reference source not ound.**.

In the microcontrollers used, it was programmed a PID control in order to control the angular velocity of the motors. With this control, the embedded PC just have to send the desired angular velocity of the motors that the microcontrollers guarantee that the motors run in the desired angular velocity.

3.3.2. IMU/GPS

The IMU used in this project is the LPMS-CU from Life Performace Research. This IMU provides the control system with roll, pitch and yaw measurements, as well as acceleration in 3 axes and angular velocity in 3 axes. This information is first sent via CAN to a Titech M4 Microcontroller developed by Hibot, which later sends the data to the embedded PC via Ethernet.

The GPS used is the Adafruit Ultimate GPS Breakout v3 with an external active antenna to improve the satellite search and guarantee a fast fix. This GPS sends the NMEA messages at a 5 Hz rate to the same Titech M4 microcontroller the IMU is sending, which after parsing the data, sends them to the embedded PC.

This system is extremely important for the boat navigation and control, once it measures the boat position and orientation. The final configuration of the IMU/GPS system is shown in Figure 8.

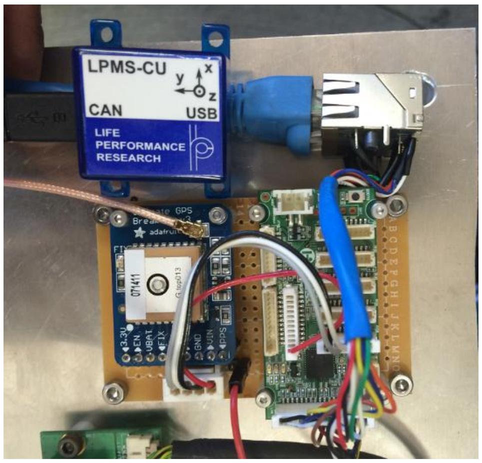

Figure 8 - IMU (Top Left), GPS (Bottom Left) and Titech M4 (Bottom Right)

3.3.3. Ultrasonic Sensor

To help in the obstacle detection of short distances, six HC-SR04 ultrasonic sensors are used in the WAM-V. Those are divided into three per ski and are disposed according to the schematics shown in Figure 9.

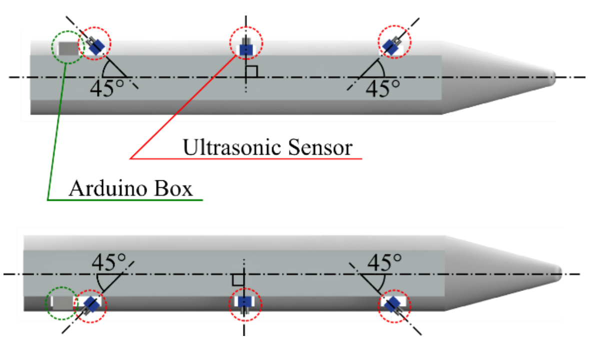

Figure 9 - Ultrasonic Sensors Disposition

The ultrasonic sensors in the middle are placed perpendicular to the skis, while the ultrasonic sensors in the front and in the back are placed 45 degrees to the skis. This configuration was chosen in order to increase the detection range.

Each of the skis has an Arduino Duemilanove attached in order to read the data from the three ultrasonic sensors and send via Ethernet to the embedded PC. It was chosen to use the Ethernet communication in order to avoid problems with communication due to the long distance between the Arduinos and the embedded PCs.

To protect the ultrasonic sensors against water, it was designed a special case for them which was manufactured in a 3D milling machine. To protect the Arduinos, it was used a small sized waterproof box from Takachi Electronics Enclosure Co. Those protection boxes with their respective content can be seen in Figure 10.

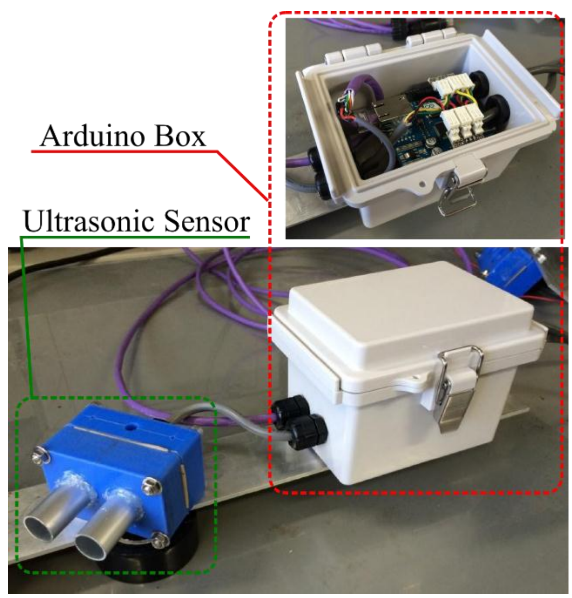

Figure 10 - Ultrasonic Sensor and Arduino Box

3.3.4. Vision System

For the Vision System, the Bumblebee® XB3 stereo vision camera from Point Grey Research is used. It is a 3-sensor multi-baseline IEEE-1394b (800Mb/s) stereo camera designed for improved flexibility and accuracy. Because of the limit of its view range, two Bumblebee® XB3 stereo vision cameras with the configuration shown in Figure 11 are used and a view range of 120 degrees could be obtained. With this configuration, two objects with a distance of 10 meters between each other can be viewed at the same time even when they are close to the camera in distance of 2.9 meters.

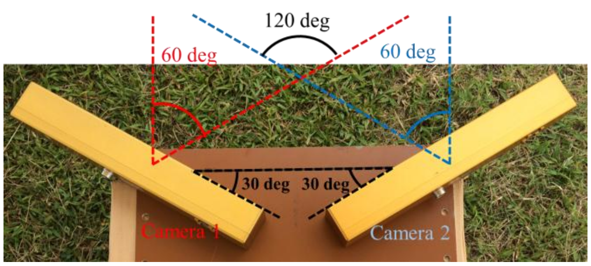

Figure 11 - Stereo Cameras Configuration

To improve the vision system capability, the IS16 Industrial LeddarTM Sensor from Leddar™ Technology is also used. This sensor is based on time-of-flight measurements using pulses from infrared LEDs, combined with 16 independent active segments. Because of the limit of its detection range, two Leddars with the configuration the configuration shown in Figure 12 are used and a detection range of 90 degrees could be obtained. With this configuration, two objects with a distance of 10 meters between each other can also be viewed at the same time as close as a distance of 5 meters.

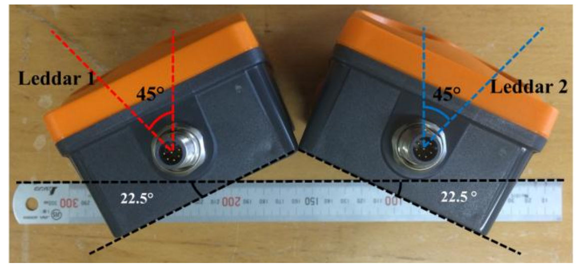

Figure 12 - Leddars Configuration

The Pantilt PTU-048 provided by sustainable robotics was used to move those cameras according to the necessity. The final configuration of the Vision System can be seen in Figure 13 .

Figure 13 - Final Configuration of the Vision System

Due to the lack of waterproof protection for the stereo cameras, a waterproof case was designed. Besides, in order to avoid distortions in the images due to the water coming for splashing or rain, an air cleaning system was placed at the front of each lens. The air cleaning system make use of a meltec mini compressor which blows compressed air directly to the frontal glass of the protection case through holes drilled in its top cover.

In Figure 14 it can be seen a test performed by the Vision System in some experiments. In those figures, it can be seen the simultaneous detection of a red buoy by camera 1 at a distance of about 6 meters and a green buoys by camera 2 at a distance of about 17 meters.

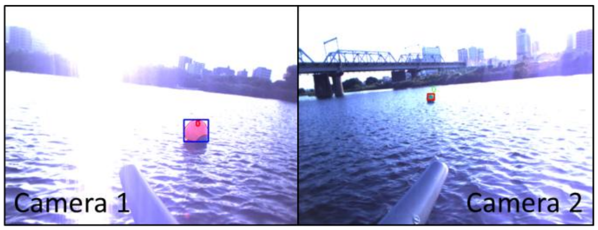

Figure 14 - Buoys Detection by Stereo Cameras

3.3.5. Pinger Detection System

To detect the pingers in Task 2, the Pinger Detection System will make use of two hydrophones from Aquarian Audio Model H2c (Figure 15).

Figure 15 - Hydrophone

Those hydrophones will be attached to the motor coupling in order to have it all the times inside the water. The data from this hydrophone will pass through 3 stages of amplifications before sending the data to an Arduino, which will treat the data and send the information to the embedded PC via serial.

3.3.6. Electronic Box

To place all the electronic components and protect them against water, it was developed what we call the electronic box. As the body of this electronic box, it was used also a water proof box from Takachi Electronics Enclosure Co. All the connections that had to be made outside the box counted with IP67 connectors which, adding the fact that the Takachi boxes are totally waterproof, guaranteed that all the electronic components were protected against water.

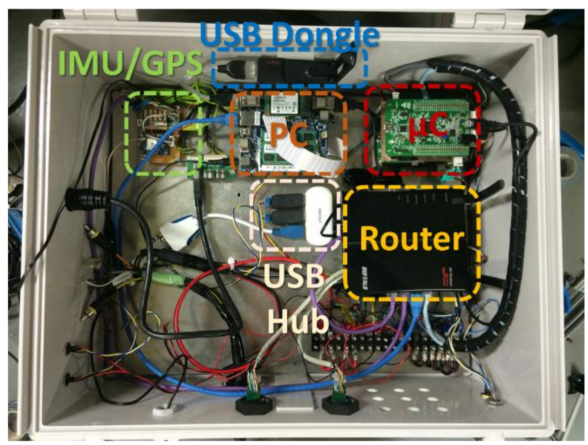

Figure 16 - Electronic Box

Inside the electronic box, it was placed also the embedded PC model GB-BXi7-4500 from Gigabyte and a Buffalo High Power Router model WHR-HP-G300N for the communication with a remote computer used by the team for monitoring. It was also placed Ethernet and USB hubs to increase the number of Ethernet and USB ports inside the box, and a Buffalo USB High

Power Dongle model WLI-UC-G300HP to communicate with the Technical Directors. The box configuration can be seen in Figure 16.

3.4. Software Approach

The software has been divided into modules that interact together for driving the boat. It is composed of 6 satellite modules and one core module (Figure 17).

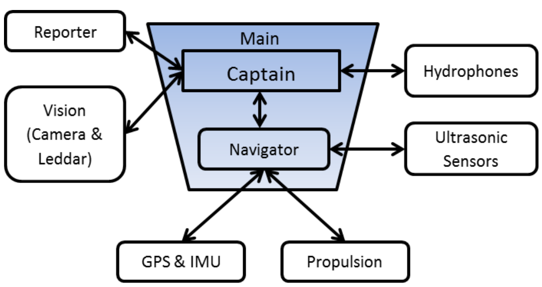

Figure 17 - Software overview

A. Satellite Modules:

- *Reporter*: In charge of maintaining contact with the judges by reporting the boat's current situation.
- *Vision*: This module gets information from the stereo-vision cameras and the Leddar sensor. Also controls a PanTilt for reorienting the sensors. Determining location of buoys, docks and 2D shapes are the main tasks of this module.
- *GPS & IMU*: This module interacts with the GPS and IMU hardware to provide location and heading information.
- *Propulsion*: This software module sends to the propulsion system's microcontroller the desired speed for each individual motor and sets the operating mode: Automatic or Manual (for remote control).
- *Ultrasonic sensors*: Provides raw distance information of the near surroundings of the boat. The ultrasonic sensors can check for obstacles in zones where the vision sensors cannot physically reach.
- *Hydrophones*: This module provides communication with the hydrophones for Task 2.

B. Main Module:

Here the Captain and the Navigator codes are located in different logical packages. However, being these two modules tightly related, both are located within the same module. The tasks strategy is embedded completely into the Captain's Module, so this act like a "mission controller". The ultrasonic sensors are directly connected to the Navigator block to act as a final fail-safe during navigation. More details on the Core Module are explained in section 3.4.2.

3.4.1. Software Modularization

We modularized our software into easily identifiable pieces through the use of robotics middleware. Our modularization objectives were:

- To develop and test each module separately without affecting other modules.
- To detect problems in an isolated way, discarding parts of the program that should not be the problem immediately, increasing debugging speed.
- To distribute the software in various devices to increasing computing power.

- To work efficiently, since there would be less code to explore and less libraries to import per module. Plus, the system integration would be easier through standard interfaces.

We decided to use RT-Middleware since it provides the required tools for us.

3.4.1.1. Integration using RT-Middleware:

RT-Middleware defines the RT-Components (RTC) as the main software abstraction [1]. RTCs are big pieces of software in charge of a particular task and has 2 types of ports:

- Data port: It can receive or send various types of data asynchronously. In our case, we used them in the early development and debugging stage of the project.
- Service port: It acts as a handler for functions. Such functions are like methods of an RTC object; they can be called from a remote machine, even if the RTC is not instantiated in the remote machine.

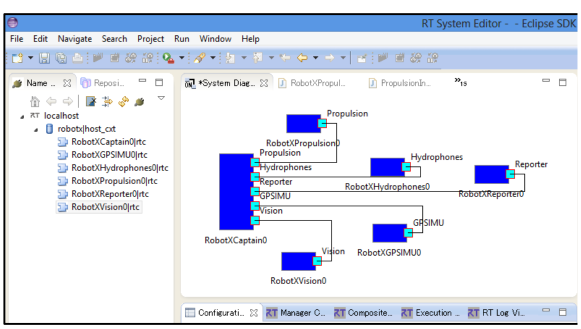

Figure 18 - System modularization with OpenRTM-aist

As a typical function call, it is a blocking call, and the code will continue running after the function has returned a response. We are using Service Ports for our RTCs, since this interface enables complex and synchronized interactions among RTCs.

3.4.2. Main module

3.4.2.1. Captain (Tasks Strategies)

The core logic for the completion of the tasks is contained within this module: it is in charge of requesting and receiving information from the sensor suite of the boat, process its current position and set goals to navigate through the tasks.

A. Captain Initialization:

Starts all components and sets the current heading as "Reference Heading". All heading calculations are made relative to the "Reference Heading".

B. Task 1 – Navigation from Gate A to B:

Figure 19 shows the task's state diagram.

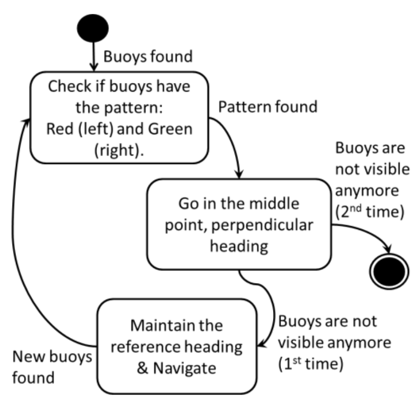

Figure 19 - State diagram for Task 1

C. Transition Task 1 to 2:

A new destination point is set to the middle of the first quadrant of Task 2. This point is calculated from the two entry GPS points of Task 2. The boat will navigate to this point and proceed with the next sequence once the point is reached.

D. Task 2 – Ping Search:

The boat will navigate to every buoy and report the quadrant and location of the buoy with maximum sensed strength signal. Figure 20 shows a simplified explanation.

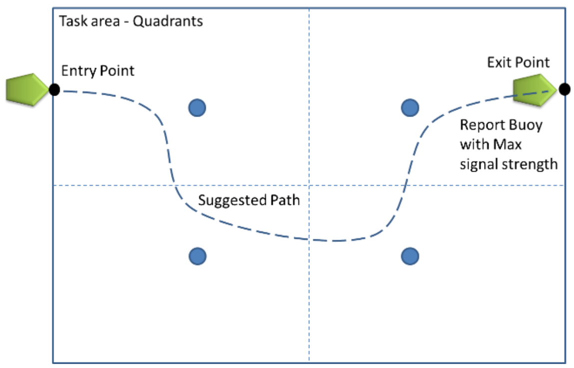

Figure 20 - Suggested trajectory for Task 2

E. Transition Task 2 to 3:

Same as in subsection C. The destination point will be set to the middle of the two entry GPS points of Task 3.

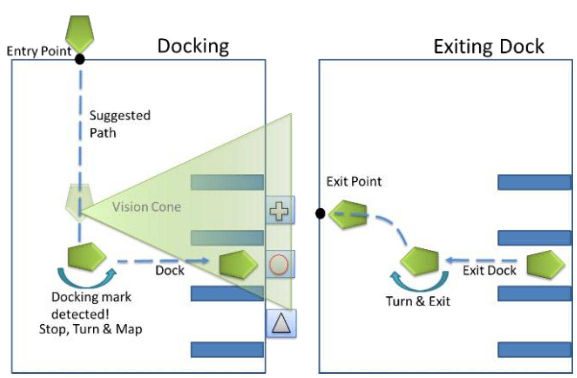

Figure 21 - Docking strategy for Task 3

F. Task 3 – Docking:

The following two diagrams describe, in a graphical way, the strategies for Docking and Exiting the Dock.

G. Transition Task 3 to 4:

Same as in subsection C. The destination point will be set to the middle of the two entry GPS points of Task 4.

H. Task 4 – Light Buoy:

Figure 22 shows the task's state diagram.

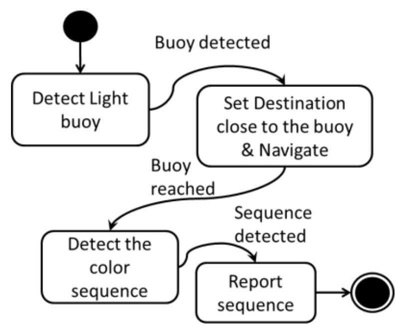

Figure 22 - State diagram for Task 4

I. Transition Task 4 to 5:

Same as in subsection C. The destination point will be set to the middle of the two entry GPS points of Task 5. The boat will navigate to this point while checking for the 4 buoys of the entry gate of task 5. When the buoys are found, the boat will proceed with the next sequence.

J. Task 5 – Obstacle Avoidance:

Being in front of the entry gates, the boat will set a destination point to the middle of the 2 buoys of its assigned entry gate.

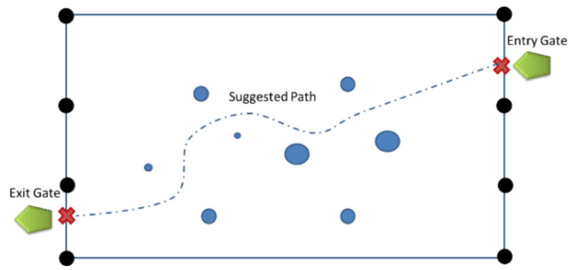

Figure 23 - Example of path for Task 5

When the gates are no longer visible, the boat will continue to head to the estimated exit gate location. In the meanwhile, the boat may encounter obstacle buoys, which are avoided generating temporary goal positions, by selecting points that are nearer the exit gate depending on the obstacle arrangement (e.g. if a single obstacle is detected, and the goal is in the left, the boat will avoid the obstacle by going to the left side of the obstacle). When the exit gate buoys are detected, the boat will set a new destination point to the middle of the designated exit gate for the run. Once the exit buoys are no longer visible, the boat will navigate forward until it reached the GPS line that marks the end of the Tasks area.

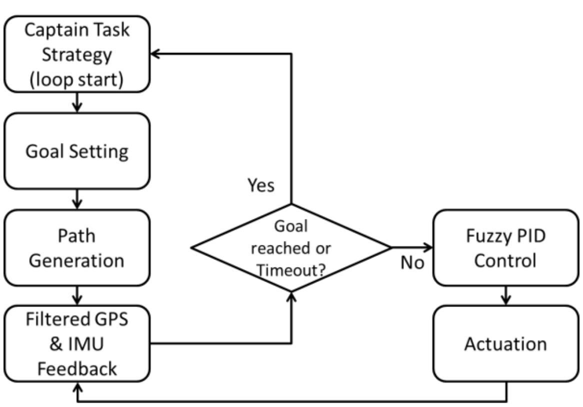

Figure 24 - Navigation flow chart

3.4.2.2. Navigator

It is in charge of driving the boat to the desired destination, set by the Captain.

A. Goal Setting:

Depending on the task, this part receives a set of obstacle coordinates, from which the goal position and goal heading is calculated (e.g. given 2 buoy positions, set the goal position to be in middle of the buoys).

B. Path Generation:

Using the defined goal, a path is generated using a Hermite spline with unit interval (scaled according to our needs) (eq. 1):

$$p(t) = (2t^3 - 3t^2 + 1)p_0 + (t^3 - 2t^2 + t)m_0 + (-2t^3 + 3t^2)p_1 + (t^3 - t^2)m_1 \quad (1)$$

Where p_0 is the starting position (i.e. current boat's X and Y position), m_0 is the starting heading, p_1 is the goal position and m_1 is the goal heading. An array of intermediate goal positions and orientations is obtained and passed to the Control algorithm.

C. Filtered Feedback:

The position feedback from the GPS has a relatively large variance, and the heading feedback from the IMU has some noise. We decided to filter those inputs.

a. GPS Position feedback:

A Kalman Filter was used to obtain filtered position and velocity data [2]. The boat system was modeled in space-state form (eq. 2). However, due to a series of time constraints in the project, we used a simpler model. To update the system state, we used the raw GPS position, and the raw IMU acceleration.

$$\begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} t^2/2 & 0 & 0 & 0 \\ 0 & t^2/2 & 0 & 0 \\ 0 & 0 & t & 0 \\ 0 & 0 & 0 & t \end{bmatrix} \begin{bmatrix} ax_{k-1} \\ ay_{k-1} \\ ax_{k-1} \\ ay_{k-1} \end{bmatrix} + E \quad (2)$$

The covariance matrices for the filter were calculated according to the datasheets of our sensors. The standard Kalman filter's "predict and update" algorithm was followed to obtain filtered position and velocity feedback for the boat's control.

b. <u>IMU Heading Feedback:</u>

In contrast with the GPS, the IMU signal was less noisy, and a "Simple Moving Average" filter was enough for this task (eq. 3).

$$\gamma_i = \frac{1}{n} \sum_{j=0}^{n-1} \varphi_{i+j} \qquad (3)$$

D. <u>Control algorithm:</u>

The algorithm is divided in three parts: Position control, Heading control and Speed control. If after a considerable amount of time the boat has not arrived at the final goal position, all the operation is suspended, the path is recalculated and the control algorithm starts all over again with the new intermediate goal positions (Figure 24). As for the position control, we determined that the boat has reached a goal by comparing the current position and desired position. As for the Heading and Speed control, we used a Fuzzy PID control [3][4][5][6] with the Triangular membership functions shown in Figure 25, where: Z means "is zero", N means "is negative", P means "is positive", D means "decrease", I means "increase" and NC means "no change". As for the defuzzification procedure, the centroid method was used.

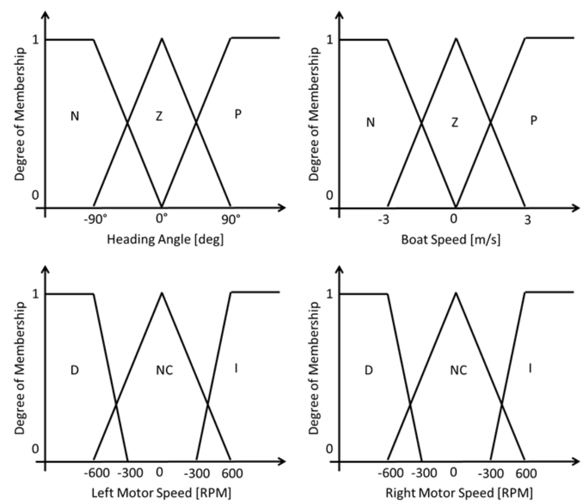

Figure 25 - Triangular membership functions for Fuzzy PID Control

a. <u>Position control:</u>

Checks in every cycle for the current position and heading of the boat, and is compared with the intermediate goal positions and headings from the generated path. If the boat is not oriented towards the intermediate goal position, an "angle to goal" is calculated and is set as the "desired heading angle". If the "distance to goal" is too large, the "speed gain" is increased, and if it is too short, is decreased. The boat is considered to be in the intermediate goal position if it is inside the “goal zone”, defined by a “tolerance radius”.

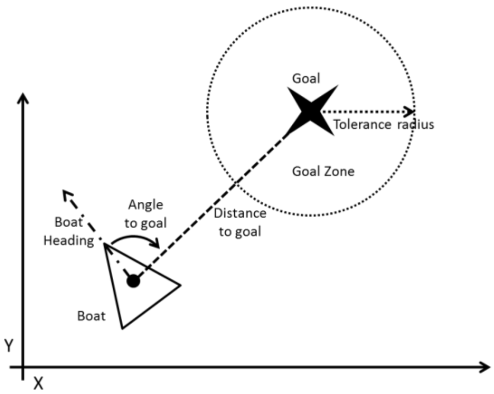

Figure 26 - Setting desired heading and speed from current heading and position

b. Heading and Speed Control:

Two Fuzzy PID controllers were used, one for heading control and one speed control. Depending on the proportional and derivative errors, output values (in RPM) for the right and left motors were calculated to drive the boat towards the desired value. For tuning each controller, 2 input gains (for proportional and derivative error) and 4 output gains (left and right motors proportional and integral output gains) were defined. The integral gains were added at the end to diminish offsets.

For heading control, RPM values were calculated only for rotating the boat towards the desired heading. In the case of the speed, RPM values were calculated only for moving the boat forward or backward to match the desired boat’s speed. The effect of the heading control was always applied to actuate motors, but the effect of the speed control was applied only if the heading error was small enough, prioritizing a correct heading.

E. Actuation:

Before applying the left and right motor actuation speeds, a last check for near obstacles is made by the Ultrasonic sensors. If there is an obstacle nearby, it will ignore the calculated values and will calculate “safe” values to rotate the boat and avoid collision until the near obstacle cannot be detected anymore. Otherwise, it applies the calculated values.

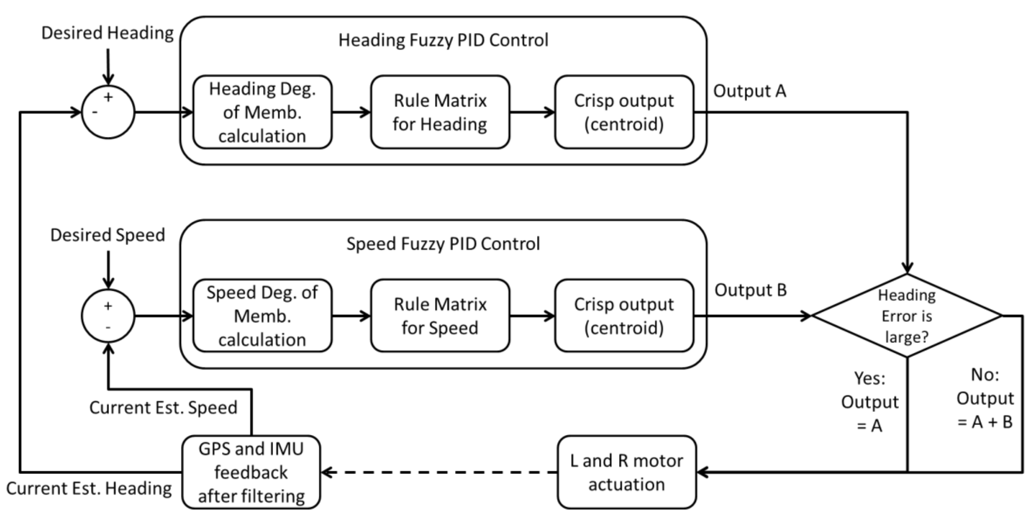

Figure 27 - Fuzzy PID control flow diagram

F. Simulation:

To test our algorithms, we developed an additional RTC to communicate with the V-REP simulator. Here, we recreated the competition place to simulate the boat's task progress and completion. This RTC acted as both the GPS-IMU RTC and the Propulsion RTC.

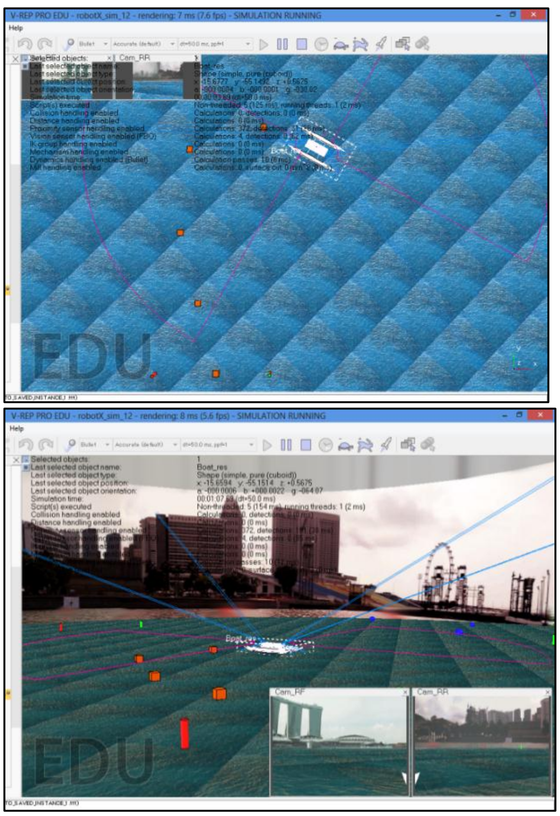

Figure 28 - Screenshots of boat simulation in V-REP (cubes represent intermediate goal positions to reach)

G. Monitoring GUI

To monitor the status of the boat, and to send configuration parameters, we are using the HyperBot GUI [7], which is an ongoing development of our laboratory that uses our previously developed Intelligent Cross-Platform Interface (ICPI) and our Framework for Integration of Elements and Resources by Roles (FIERRo). In this GUI, The boat status can be visualized and the configuration parameters can be set through a web browser. The communication protocol is Websocket.

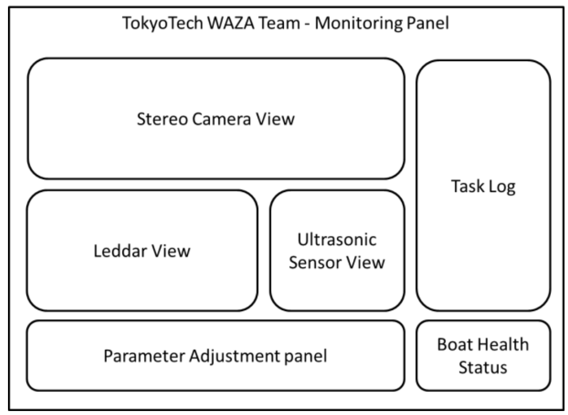

Figure 19. Monitoring GUI layout

4. Collaboration

To make this development possible we received the collaboration of our advisor's laboratory, the Fukushima Robotics Laboratory, and three other companies than the competition sponsors which are Sustainable Robotics, Tokyo Aburagumisou Honten and Gyomu Super Japan Dream Foundation.

The Fukushima Robotics Laboratory provided us with most part of the sensors, microcontrollers and embedded PCs needed for the development of the electronic system of our robot. It also provided us with a space for the development of our boat and to hold meetings to discuss the project, which count always with the participation and advice of Prof. Fukushima.

Sustainable Robotics provided us with a PanTilt which was used for the vision system and the Tokyo Aburagumisou Honten provided us with some money for the development.

The Gyomu Super Japan Dream Foundation, after we passed in their selective process, provided us with a large amount of money which helped us covering most part of our travel, shipping and development expenses.

To obtain all those sponsorships, it took us a lot of negotiation and time. Therefore, what we have learned by this experience is that obtaining sponsorships should be one the first things we have to do when starting a project, since company support opens the possibilities of development which we realized it was quite limited at the beginning of the project because of the lack of it. We also learned that sponsorships should be obtained before the end of the fiscal year once after that, it is difficult for companies to provide us financial support here in Japan.

5. Conclusion

After one year working in this project, we can conclude that we achieved the goal of developing an autonomous USV capable of complete tasks based on marine applications. However, we cannot conclude that it was an easy task since we had to face several issues at early stages such as the lack of financial resources due to the delay in the reception of the stipend (4-5 months delay) and lack of time and human resources due to the research and graduation of some members. This facts for sure constrained our project, but we still were able to fulfil our goals.

With this project we learned how to develop bigger scale robots, a thing that not all the team members had the opportunity to deal with. We also learned how administrative and financial matters are important for the development, and how fast this should be deal with in order not to affect the project development.

As a future work we would like to continue developing USVs to use for other applications such as search and rescue, hazmat spills monitoring (e.g. in Fukushima Prefecture area) and tsunami detection and warning. We would also like to continue developing it to use as a platform to release other small AUVs for surveillance or monitoring of marine life.

6. References

[1] Ando, N.; Suehiro, T.; Kitagaki, K.; Kotoku, T. and Woo-Keun Yoon: RT-middleware: distributed component middleware for RT (robot technology), 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3933-3938 (2005)

[2] Gomez-Gil, J.; Ruiz-Gonzalez, R.; Alonso-Garcia, S. and Gomez-Gil, F.: A Kalman Filter Implementation for Precision Improvement

in Low-Cost GPS Positioning of Tractors, Sensors, Vol. 13 pp. 15307 - 15323 (2013)

[3] El-Khatib, M. and Hamilton, D.: A Layered Fuzzy Controller For Nonholonomic Car-Like Robot Motion Planning, 2006 IEEE International Conference on Mechatronics, pp. 194- 198 (2006)

[4] Rashid, R.; Elamvazuthi, I.; Begam, M. and Arrofiq, M.: Fuzzy-based Navigation and Control of a Non-Holonomic Mobile Robot, Journal of Computing, Vol. 2 pp. 130- 137 (2010)

[5] Vaneck, T.: Fuzzy guidance controller for an autonomous boat, IEEE Control Systems, Vol. 17 N. 2 pp. 43- 51 (1997)

[6] Sang-Min, Lee, Kyung-Yub Kwon and Joongseon Joh: A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines, International Journal of Control, Automation, and Systems, Vol. 2 N. 2 pp. 171- 181 (2004)

[7] Ceron-Lopez, A. and Fukushima, E.: Basic Study on Element Administration in Robotics Middleware – 3rd Report: A GUI based on “role definitions” for using elements and resources in robotics software, Proc. of the 2014 Robotics Society of Japan Conference, 2A1-05 (2014)