SAUC-E 2012 - UNIFI Team - Turtle vehicle

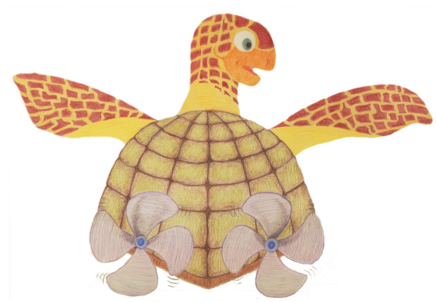
F. Bartolini[*], R. Costanzi[*], N. Monni[*], A. Ridolfi[*], G. Vettori[*], F. Giardi, E. Lupi, E. Mazzuoli, M. Montagni, L. Paolucci and B. Allotta[*]

June 20, 2012

[*] MDM Lab, Mechatronics and Dynamic Modelling Laboratory
Energy Engineering Department
University of Florence
via S. Marta, 3, 50139 Florence, Italy
a.ridolfi@unifi.it

Abstract

The paper describes the University of Florence (UNIFI) vehicle designed to participate to the 2012 SAUC-E competition. The UNIFI vehicle name is TURTLE. The PhD students and the students of the MDM Lab (Mechatronics and Dynamic Modelling Laboratory) of the University of Florence have designed Turtle during the last 8 months. Turtle has been built both using commercially available products and constructing some parts inside the MDM Lab (e.g. structural components and magnetic transmission using 3D rapid prototyping machine). This is the first time the UNIFI Team attends to the SAUC-E competition. That's why the team decided to face only some of the tasks proposed by the NURC organizers.



1 Introduction

The team of the University of Florence (UNIFI) has decided to attend for the first time to the SAUC-E 2012 competition. Turtle vehicle, that has been built for the SAUC-E 2012 competition, has been designed and developed inside the MDM Lab, the Mechatronics and Dynamic Modelling Laboratory of the University of Florence. Over the past two years, the MDM

Lab has acquired experience in Marine Engineering field and in particular in the underwater robotics, thanks to several collaborations in national and international research projects. The MDM Lab has been partner of the THESAURUS project, funded by Regione Toscana, where has developed a class of modular Autonomous Underwater Vehicles (AUV) able to cooperate in swarms, in order to perform the navigation, exploration and surveillance of the underwater archaeological sites. In May 2012 the MDM Lab was admitted as coordinator of the three-year European project ARROWS (ARchaeological RObot systems for the World's Seas). In that project innovative robotic technologies for archaeological investigation of very diversified environmental conditions are proposed, with testing sites in the Mediterranean Sea (Sicily) and Baltic (Estonia). The interest of many students, coming from different study paths, for the design of our vehicle and its control strategy has made possible the development of Turtle vehicle, specifically for the SAUC-E 2012 event. Inside the development of this project bachelor as well as master thesis have been done.
All the sensors are interfaced to an embedded PC104 module, where the control software of the robot runs with the Ubuntu-Linux OS. A modular software architecture has been developed through the open source MOOS-ivp suite. The code is written in C++ and exploits OpenCV libraries for the image processing. For the control of the motors and of the strobe light an Arduino board has been used. The structure of this paper is the following one: in section 2 Turtle hardware is described, as concerns its mechanical part, its electronics architecture and the used sensors; in section 3 Turtle software is described, as regards the sensor data acquisition, the control system and the vision data processing; in section 4 the team describes how to face the chosen tasks (in particular the validation one, the underwater structure inspection and the wall survey).

2 Turtle Hardware

2.1 Mechanical characteristics

Turtle vehicle is made by a central case (Figure 1) in order to achieve the following aims: containing the instrumental hardware and ensure the necessary buoyancy, according to the competition rules.

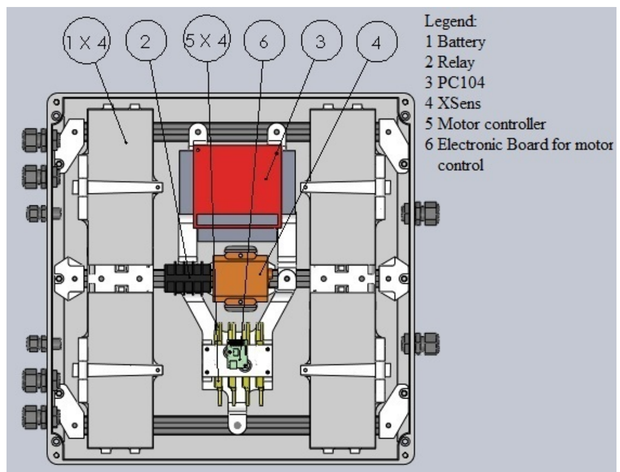


Figure 1: Central case of Turtle

The case is placed inside a frame made of aluminum profiles. The motion of the AUV is given by 4 propellers:

- two vertical ones, necessary to control the depth and the pitch angles;
- two horizontal ones, for the forward thrust and the advancing direction control.

The echo-sounder sensor is placed on the bow of the vehicle. The webcam is placed in a dedicated watertight case located under the vehicle. Inside the other watertight case there are the depth sensor and the strobe light. In Figure 2 and 3 the whole structure of Turtle and its total dimensions are shown. The total weight of Turtle vehicle is about 27 kg.

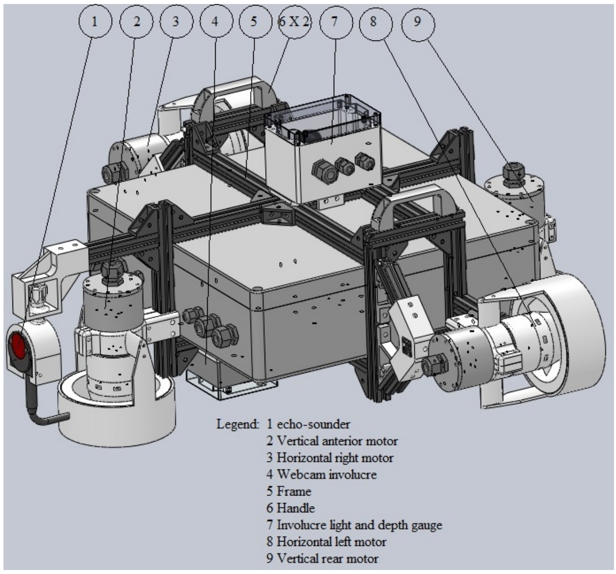


Figure 2: Mechanical structure of Turtle

2.2 Electronics architecture

2.2.1 Power supply

Turtle vehicle is equipped with four 22.2V LiPo batteries; each of them can deliver 8Ah and supplies one of the four electrical motors. The PC104, the Echo Sounder, the Arduino board, the wireless receiver and the strobe light supply is evenly distributed among these batteries. The power supply of the other devices, such as the Xsens and the two webcams, is provided by the PC104 through an USB connection. The four batteries can be disconnected through a contact controlled by a magnetic switch. This means it is possible to turn on and off the AUV just touching the hull with a magnet. Figure 4 shows the simplified scheme of the power and control lines of the vehicle.

2.2.2 Device Control

Some of the AUV devices cannot be directly connected to the PC104 module. The Arduino board controls the pressure sensor (depth sensor) and the strobe lights and works as an interface between the PC and these devices. It communicates with the PC through a serial COM port. Furthermore it performs

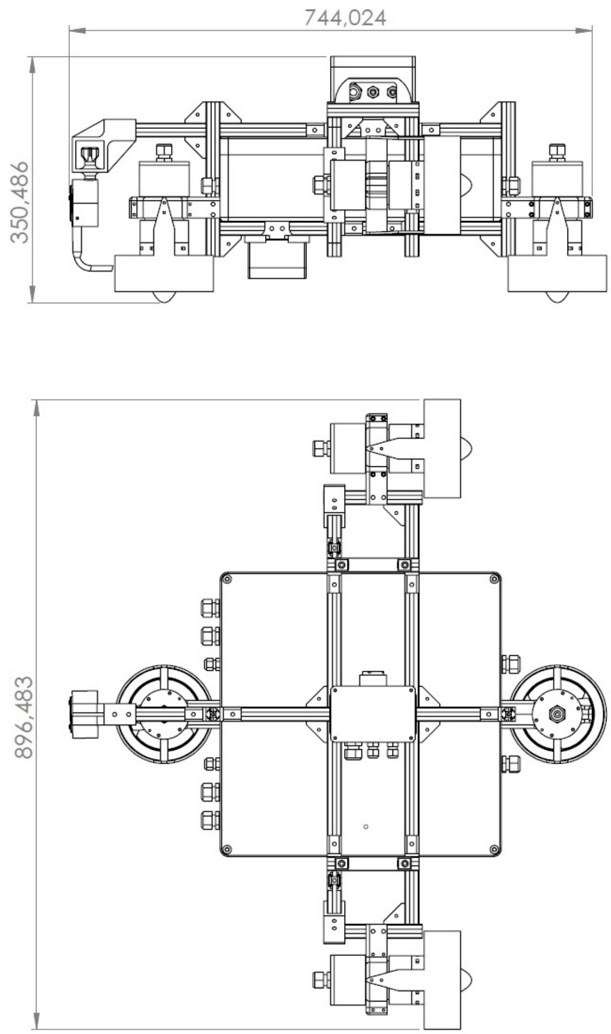


Figure 3: Turtle sizes

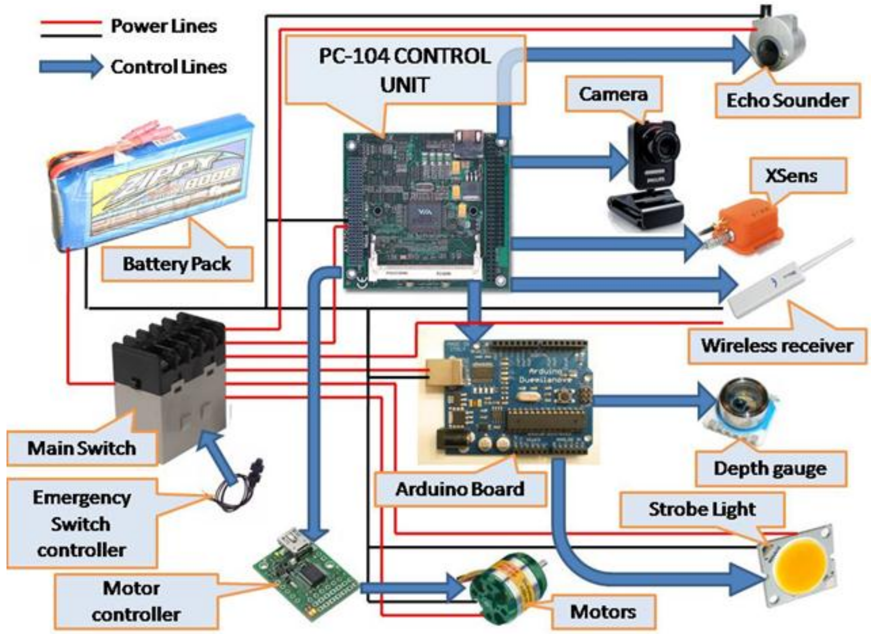


Figure 4: Power and control schema

a measurement of the level of charge of the batteries and controls the switching of the strobe. The motor control is performed through the Micro Maestro control board.

2.2.3 Propelling system

The AUV has 4 electric Robbe motors which usually are employed for hobby purposes. The chosen model is the ROXXY BL-Outrunner 3548/06. Despite its small dimensions and low weight this motor provides quite a high torque. Each motor is controlled by a ROXXY BL 940-6 controller and powered by a 22.2V LiPo battery. The control signal is provided to each controller by the Micro Maestro control board which converts the PC output into the motor command.

2.2.4 Strobe lights

According to the 2012 competition rules, Turtle must be equipped with a strobe light. This device has a double purpose: showing the AUV position in the pool and, when the battery level goes down, warning about it using a different pulse rate of its light. The switching rate is set by the Arduino board since it monitors the battery level.

2.3 Sensors

The choice of sensors installed on board has been determined on the basis of the tasks the team has chosen to perform this first year.

2.3.1 AHRS - Xsens MTi

The MTi Xsens contains 3D gyroscopes, accelerometers and magnetometer. The internal low-power digital signal processor runs a real-time proprietary sensor fusion algorithm providing drift-free 3D orientation data. The sensor datasheet ensures a RMS dynamic accuracy of 2 degrees. Additionally, 3D dynamic data are given by the gyroscopes, accelerometers and magnetometer, so that it will be possible during experimental phases to compare performances of the proprietary algorithm with the algorithms implemented by the team. This will allow to verify the validity of the results of simulations carried out by the team. The sensor is plugged to PC104 through a USB port for both data exchange and power supply.

An on site calibration procedure will be necessary for the characterization of the local magnetic field disturbances and, compensating them, to work properly. The MTi Xsens is already equipped with a software tool for this procedure.

2.3.2 Sensor pressure - MS5535-CM

To measure the local pressure, the MS5535-CM sensor from Measurement Specialties Inc. has been chosen. The MS5535-CM is a 16bit digital sensor with an operating range up to 14 bar. The sensor is soldered on a little board built by the team and drowned in the resin with the board itself to isolate all the electric contacts from the water. The sensor communicates through a serial protocol with the Arduino board, that then provides data to PC104 via USB. The sensor is also equipped with a temperature sensor to compensate the pressure measurement error.

2.3.3 Webcam - Philips SPZ5000

One webcam is mounted in a watertight box, with transparent cover, placed under the vehicle. The camera is the SPZ5000 by PHILIPS, chosen especially for its brightness (maximum aperture is F:2.6), for its wide angle (water will reduce it) of 80 degrees in air and for its resolution of 1.3 MP. The camera is plugged to PC104 through a USB port for both data transmission and power supply.

2.3.4 Echo Sounder - Imagenex 852

An Imagenex 852 is mounted on the front part of the vehicle in order to measure the distance of the vehicle from obstacles. The device is adjustable around a vertical axis to set the working direction on a horizontal plane. This will be useful in the wall survey task where the distance (>2 m) to be maintained between the vehicle and the wall is enough to make impossible the use of cameras. The sensor is characterized by a conical transducer with a 10 beam width, a 8bit resolution and an adjustable maximum range between 5 m and 50 m. The sensor is plugged to the PC104 via RS232 port, whereas the power is supplied by the battery pack (22-30 VDC required).

3 Turtle Software

3.1 Software organization

The software part is organized in order to provide a reliable structure which makes easy and fast the code testing. The guidance of the vehicle is composed of many processes such as the acquisition of the outputs of the devices, such as the Xsens platform, the depth sensor, etc. and the control of the thrusters. A program managing the communication among all the processes of the software is very useful considering that different people work on many tasks to be linked together. For this reason the MOOS-ivp system [1] has been chosen, where MOOS stands for "Mission Oriented Operating Suite". On this platform every process is encapsulated in a special MOOS class and works as a client-process; a main process called MOOSDB works like a server, managing a database and handling the communication among the processes: every client-process asks MOOSDB the access for the publish/subscribe activity to the database. So every developer can test his process with MOOSDB with the reasonable certainty the process will be compatible with the others. The scheduling of the processes is handled by the operative system (in this case Ubuntu Linux) that also provides the managing of any low level interface to the hardware; this scheme is an easy and full-tested way to organize the process communication. Figure 5 shows the architecture of the client-processes, managed by the MOOSDB.

[1] http://www.robots.ox.ac.uk/~mobile/MOOS/wiki/pmwiki.php, oceanai.mit.edu/moos-ivp

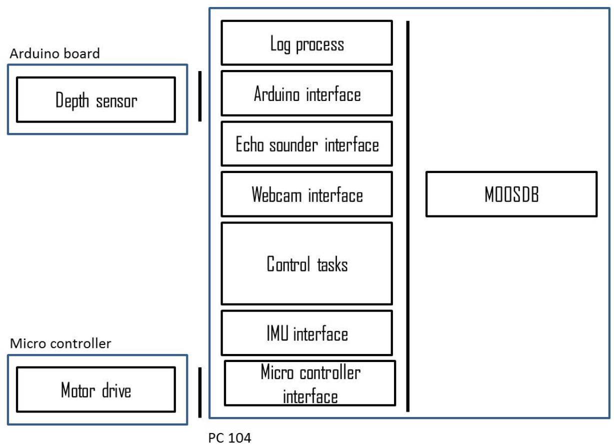


Figure 5: Software scheme

3.2 Architecture of Control

The architecture of control has been designed as a set of finite-state machines corresponding to the number of missions to be performed. The structure for each state machine is similar: a supervisor task manages the submissions, monitoring the occurrence of some events and, consequentially, enabling or disabling the tasks which implement some behaviours. The logic of the supervisor is related to the specific mission, as well as the number and the content of the behaviour tasks, called submission tasks. In a preliminary phase, the number of the mission can be set in order to load all the tasks related to the chosen mission. Some tasks, in fact, are shared among the different missions, such as the communication tasks, the log task and some other utilities.

The architecture control for a generic mission is shown in Figure 6.

The core of the architecture is the Supervisor task which drives the state machine on the basis of the monitored events: for instance, when through the communication task, the "start" of the mission is provided, the supervisor enables the first submission task which, for all the missions, corresponds to a submersion phase (likewise the last is always a surface phase). Other events, which the supervisor can exploit to change the state of the control, are given by the submission tasks themselves (for example the value of the tracking error of a variable) or by some utility tasks which manage the image process information and enable events such as the entrance of the pipe on the screen. The number and the nature of the submission tasks are related to the specific mission loaded, but all of them share the high level control task which implements the kinematic control described in section 4 and which is fed by the sensor measurements. The submission tasks provide their controller output to a task called Maestro (by the name of the Micro Maestro motor controller) which has the aim to add the compensation contribution given by the WB Compensator task and to distribute the forces and the moments between the four motors. The information which is relevant to be monitored is sent to the Log task and to the Communication task.

3.3 Image processing

The open source OpenCv libraries and the framework Qt4 have been chosen to implement the computer vision algorithms. The choice of "custom" algorithms has been made to maximize the information coming from the webcam.

Assuming that the colour of the pipe is in strong contrast with the background, we decided to use HSV space colour that allowed us to wisely select a single reference hue (Figure 7). The HSV colour space allows a greater selectivity of individual colours, even if they do not belong to the primary colours RGB (Red, Green, Blue).

To easily process the acquired images these are converted into binary ones: pipe and no pipe pixels. The colours different from the selected hue have been replaced with black, then the figure has been converted into a grey-scale image and binarized through the OTSU method, in order to automatically perform the image thresholding. The OTSU method is to be preferred over a manual approach since it allows a better noise minimization.

When the edges of the pipe are both in the image it is possible to extract the following information:

- the angle between the pipe and the vertical of the camera (Figure 8),

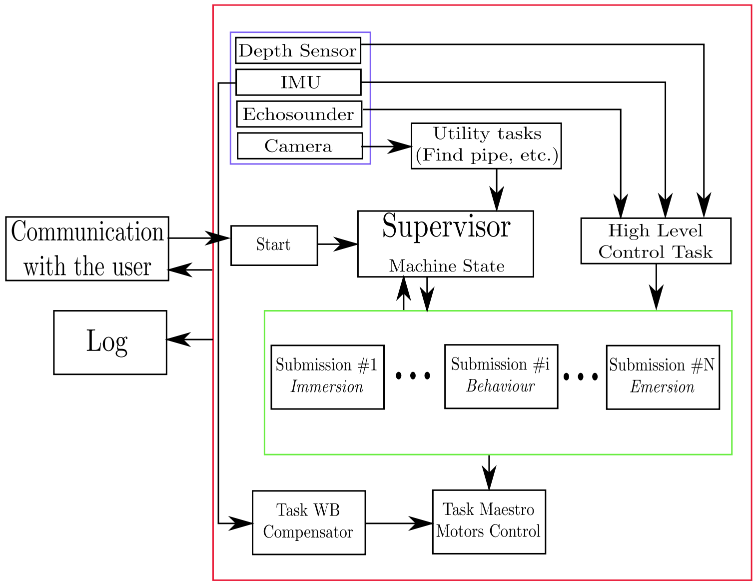


Figure 6: Architecture of control

- the barycenter of a pipe portion entering the image (Figure 8),

- the side barycenters for the recognition of the entering side of the pipe in the picture.

The angle of the pipe is calculated from the slope of the linear regression of the row midpoints (of the bigger white cluster) according to the following expression:

$$\begin{cases} y = mx + q \\ m = \dfrac{N\Sigma\left(x_i y_i\right) - \Sigma\left(x_i\right)\Sigma\left(y_i\right)}{N\Sigma\left(x_i^2\right) - \left(\Sigma x_i\right)^2} \\ q = \dfrac{\Sigma\left(y_i\right)\Sigma\left(x_i^2\right) - \Sigma\left(x_i\right)\Sigma\left(x_i y_i\right)}{N\Sigma\left(x_i^2\right) - \left(\Sigma x_i\right)^2} \end{cases} \tag{1}$$

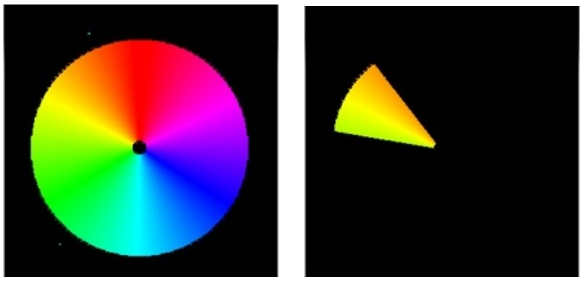
Since the results of the above expression are useless if the image is too noisy (Figure 9) a standard deviation index of the middle points of the line is associated with the image, according to:

$$\sigma = \sqrt{\frac{\Sigma\left(x_i - \overline{x}_i\right)^2}{N-1}} \tag{2}$$

where x_i is the x-coordinate of the line in the i-th row, and $\overline{x}_i$ represents the middle point. The barycenter of the pipe (identified by a cluster of white pixels) is useful to get the direction to be followed when it is not completely within the image and the index of a straight angle is not reliable. Similarly, the lateral barycenter can be used to estimate the direction of the entrance of the pipe into the image. The upper and the lower barycenters, together with the angle of linear regression allow to track the pipe, even if the image is too noisy.

4 Innovation

In this paragraph the main innovation features of Turtle are described; the innovative characteristics have been introduced by UNIFI students. Some sup-



(a) (b)

Figure 7: (a) Full colour spectrum. (b) The same spectrum after the yellow filtering.

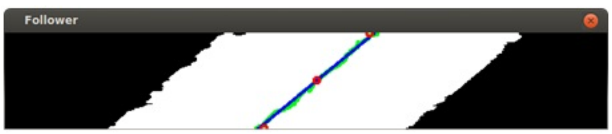


Figure 8: The figure shows the middle points of the white pixels (dots in green), the linear regression (blue) and the central, upper and lower barycentres (red).

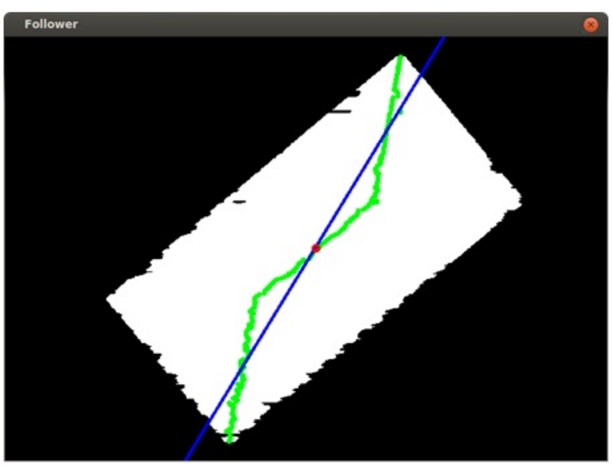


Figure 9: The figure shows the midpoints (green), the barycentre and the linear regression (blue) with a very high standard deviation (mid points are far away from the linear regression).

ports, the motor transmissions and the propellers have been designed on our own and have been built inside the MDM Lab (Pistoia, Italy) using our 3D printer (a rapid prototyping machine).

As regards the power transmission from each motor to the corresponding propeller the team decided to use a magnetic coupling; the torque transmission is given by the action between the magnetic poles of the two parts of the magnetic joint. The advantages of our solution, the magnetic transmission, are:

- it is easy to obtain a watertight motor case;
- the maximum transmissible torque is limited by the magnetic coupling itself.

On the other hand the disadvantages:

- quite high weight;
- magnetic noise on the vehicle instrumentation;
- short delay in the torque transmission between the motor and the propeller.

COMSOL simulation software has been used to design the magnetic transmission, Figure 10. The radial magnetic joint able to transmit the desired torque has 18 poles with an air gap of 3.5 mm.

After the design phase the magnetic joint has been printed with our 3D prototyping machine (Figure 11).

We used a dedicated software in order to design the propellers. The blade profiles have been drawn starting from the desired thrust as a function of the motor angular speed. Turtle propellers are made by the 3D printer and then tested in the swimming pool of the MDM Lab.

5 Motion Control

5.1 Test 1: validation

The goal of this task (SAUC-E Mission and Rules, Task 1) is to show that the vehicle can proceed in a straight line from a starting point, maintaining a

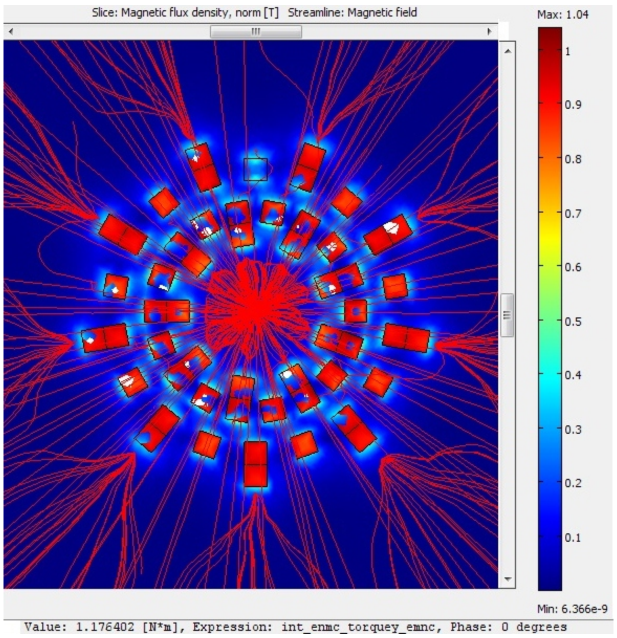


Figure 10: Design of the magnetic joint

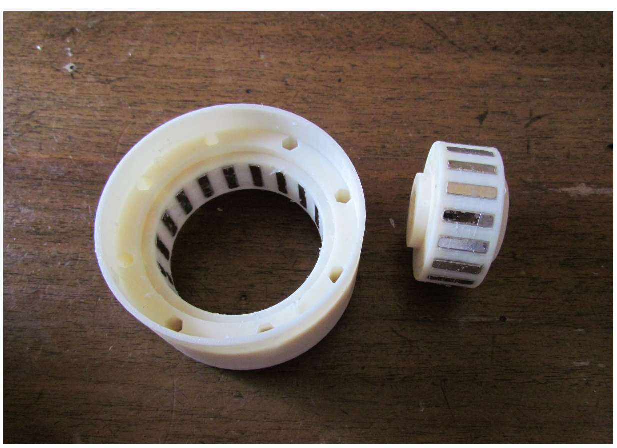


Figure 11: Magnetic joint

Figure 12: One of Turtle propellers

constant depth, and, when the pipe is reached, turn on the spot for 90 degrees and continue straight to pass through the validation gate.
The control strategy chosen to complete this task is called *waypoint navigation*, a technique based on a trajectory given by a broken line connecting points where the vehicle stops to change its orientation. The vehicle navigation exploits on-board sensor measurements:

- orientation and angular velocity, using the Xsens sensor;
- depth estimation, given by the pressure sensor;
- speed information, useful to understand if the vehicle is stationary or if it is moving with respect to an inertial frame, from camera image processing.

The motion strategy is the following one:

- once submerged till the desired depth, the vehicle proceeds from the start point with an open-loop advancing thrust profile and a closed-loop orientation control; the latter is obtained with a simple PID regulator with a feed-forward compensation of buoyancy and gravity effects. Advancing thrust profile is thought to a vehicle speed initially quite high, about 0.3 m/s, and a lower one (0.1 m/s) when it is expected to be near the pipe. PID regulator controls yaw and pitch angles with respect to a desired reference: for yaw angle it is a constant zero value, for pitch angle it can be deduced from a high level cinematic controller. Briefly, its value is chosen to correct vehicle depth using advancing thrusters;

- when the camera approaches the pipe (this event can be recognized by constantly monitoring the yellow pixel percentage in the captured image) a reference distance can be used within a PID regulator in order to place the vehicle over the desired trajectory corner point;

- when the vehicle is positioned approximately stationary over the pipe, it is controlled to rotate around its vertical axis to line its bow up to the validation gate. This rotation follows a trapezoidal angular velocity profile;

- when the vehicle angular position is correct, it can advance again with an open-loop thrust profile and a closed-loop orientation control, as described in the first point.

This control procedure has been preliminary tested in the simulation environment Matlab-Simulink^{TM}. The results are shown in Figure 13: according to that, the vehicle centers the middle of the gate and maintains its depth quite well.

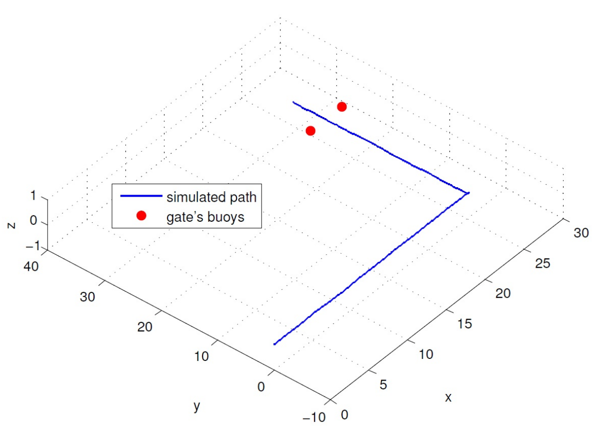


Figure 13: Validation test: simulation results

5.2 Test 2: the underwater structure inspection

The second test aim (SAUC-E Mission and Rules, Task 2) is to perform an underwater structure inspection. For this task it has been chosen a control strategy using the bottom camera output, the orientation estimated by the Xsens platform and the depth sensor measurement. The proposed approach uses a so-called *path-following* control strategy, equally suitable for sea, land and air vehicles. Traditionally, guidance-based path following strategy emphasizes spatial localization as a primary objective, and considers temporal requirements as secondary objectives, sacrificable if necessary. Classical path-following tasks are thus:

- Geometric Task (primary): make the position of the vehicle converge to and follow a desired path;

- Dynamic Task (secondary): make the speed of the vehicle converge to and follow a desired speed assignment.

The motion strategy is built to attempt the geometrical task and to maintain a constant speed as a secondary aim. In order to comply with the first task a PID regulator is used: the geometric task is achieved defining two tracking angles (one in the vertical plane and the other in the horizontal one). This kind of control strategy has been implemented in Matlab-Simulink^{TM} environment. The results are shown in Figure 14; the maximum deviation from the desired path is about 20 cm along edges. This is a very encouraging result, considering that the advancing simulated speed is quite high (about 0.3 m/s).

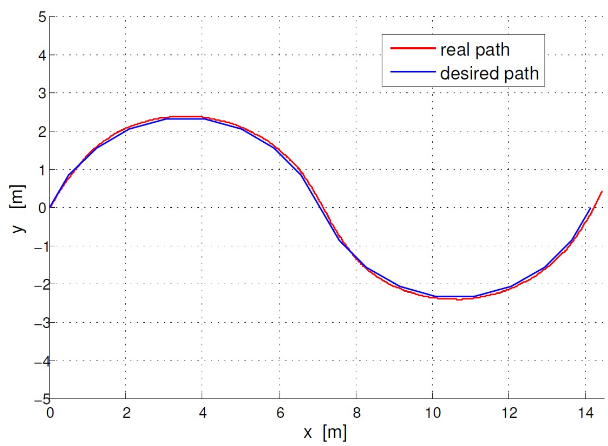


Figure 14: Pipe following task simulation results

5.3 Test 3: survey of the corner of the wall

The third task the team has chosen to face is to survey a corner of the wall of the NURC's sea basin (SAUC-E Mission and Rules, Task 4). Particularly, the aim of this task is to maintain a distance between 2 and 4 meters from a not straight wall. The vehicle is equipped with an echo sounder for the attainment of this test. This sensor is used to measure the distance to the target inside its vision range. The output of the echo sounder is an array of 256 elements: each element can assume a value between 0 and 128. The i-th of the 256 values represents the intensity of the echo coming from a point placed at a distance of $i/256$ of the sensor full scale, adjustable between 5 and 50 meters, in front of the sensor. Moreover, the sensor provides a measure, based on a data thresholding procedure, of the distance of the nearest object. A checking algorithm has been developed by the team to process the output in order to differentiate the objects that may represent obstacles, from the others that represent noise. A thresholding value of intensity is imposed to the elements of the array to recognize the closest thing of interest, which might be an obstacle for the AUV. The echo sounder uses a conical spectrum of vision with a range of $\pi/18$ rad approximately. On the basis of the procedure of data acquisition, simulation tests have been designed using Matlab-Simulink^{TM}. The idea behind the strategy of this task is the following one: whenever the AUV measures, by the echo sounder, a distance different from the desired one, the error is calculated and fixed by means of a torque rotating the AUV. As the vehicle is under actuated (sway and roll degrees of freedom cannot be actuated), it has been decided to operate with rotations around the vertical axis to fix lateral distance error to the wall. The surge thrust is set to a constant value, by means of an open-loop control strategy. The depth is controlled by the same algorithm used in the other two previous tasks.

6 Conclusions

The name we chose for our vehicle, Turtle, is not random: this is the first time the UNIFI team faces the SAUC-E competition and our goal is not to win, but to have some satisfactory results and to put the foundations for the next years. The starting know-how of our team is based on previous underwater projects developed by the UNIFI MDM Lab, such as THESAURUS. In fact some of the technologies used for Turtle development are coming directly from that project. For this year we want to focus our attention just on few tasks: the qualification one and just two others, in order to reduce the complexity of our vehicle and to try to get good results just in few things. The UNIFI team is very motivated and happy to compete on this event.

7 Acknowledgments

The UNIFI team wants to thank other people who helped us in Turtle development during these months: in particular, Luca Pugi (UNIFI Researcher) and Marco Natalini (UNIFI consultant) as well as Filippo Cangioli, Alessio Fusco, Gino Geri, Stefano Gherardini, Leonardo Lombardi and Marco Pagliai (UNIFI students). We also want to thank Francesca Miotto, the designer of Turtle logo.

References

[1] Fossen T., *Guidance and Control of Ocean Vehicles*, I Ed., John Wiley & Sons, Chichester UK, 1994.

[2] Sciavicco L., Siciliano B., *Robotica industriale Modellistica e controllo di manipolatori*, McGraw-Hill Libri Italia, I, 2nd Edition, Milano, 2000.

[3] Breivik M., Fossen T. I., *Guidance-Based Path Following for Autonomous Underwater Vehicles*, in Proceedings of the OCEANS'05, Washington D.C., USA, 2005.

[4] Carlton J., *Marine Propellers and Propulsion*, 2nd edition, Elsevier, 2007.

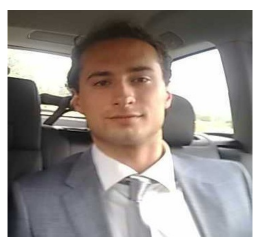
[5] Pivano L., Johansen T. A., and Smogeli . N., *A Four-Quadrant Thrust Estimation Scheme for Marine Propellers: Theory and Experiments*, IEEE Trans. on Control Systems Technology, Vol. 17, NO. 1, 2009.

8 The UNIFI team

Fabio Bartolini is one of the academic supervisors of the team. He has his master degree in Mechanical Engineering and he is a Researcher at the Dept. of Energy Engineering, University of Florence. His work lies mainly in the design and control of mechanical rig/device. Inside the team, his responsibilities are all the things related to the vehicle design, maintenance and further development.

Riccardo Costanzi is one of the academic supervisors of the team. He took his master degree in Automation Engineering last year, with a thesis on the localization of AUVs for the THESAURUS project. Currently he is a Ph.D student at the Dept. of Energy Engineering, University of Florence, where he is studying Mechatronics and Robotics, mostly as concerns Underwater Robotics. He has worked both on mechanical and control Turtle vehicle issues.

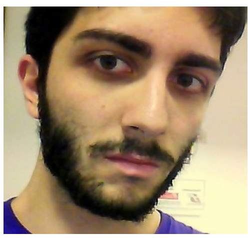
Niccolò Monni is one of the academic supervisors of the team. He has taken his master degree in Automation Engineering this year, with a thesis on AUV navigation inside the THESAURUS project. Currently he is working at the Dept. of Energy Engineering, University of Florence. He has worked on the vehicle navigation and on Turtle software part.



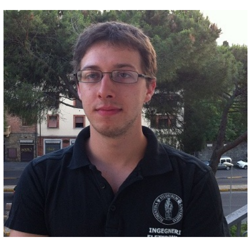
Alessandro Ridolfi is the project team leader. He has his master degree in Mechanical Engineering and he is studying as a Ph.D student at the Dept. of Energy Engineering, University of Florence. His interests lie in Robotics, vehicle dynamics and control. He has coordinated the team, dealing with technical issues and the project management.

Gregorio Vettori is one of the academic supervisors of the team. He has his master degree in Automation Engineering and he is a Ph.D student at the Dept. of Energy Engineering, University of Florence. His interests lie in vehicle navigation and software architecture. He has worked on the software implementation of the control and of the low-level drivers.

Francesca Giardi is a student at the University of Florence and she is currently working on her Master Thesis in Mechanical Engineering. Her interests are located in mechanical system control, mobile robots and Autonomous Underwater Vehicles since last year. She has worked on the AUV dynamic model and control strategy.



Emilio Lupi is a student at the University of Florence and he is currently working on his Master Thesis in Electric and Automation Engineering. His interest is focused on Robotics, in particular on inertial system and autonomous robots. As regards the SAUC-E competition, he has developed the IMU interface acquisition and the log process.



Elia Mazzuoli is a student at the University of Florence and he is currently studying for his Master Thesis in Electric and Automation Engineering. He deals, for years, with programming (both for passion and work). Inside the UNIFI team Elia deals with the computer vision.

Marco Montagni is a student at the University of Florence and he is currently studying for his Master Thesis in Electronic Engineering. In his first level thesis, about the THESAURUS project, he worked on the power system and made the watertight engine tests. Because of the experience he has gained during this activity he has worked on Turtle propelling system.

Libero Paolucci is a student at the University of Florence and he is studying for his Master Thesis in Electric and Automation Engineering. His main responsibility is the design of Turtle supplying system. He has also worked on the motor control, on the depth sensor and on the strobe lights.

Benedetto Allotta is the academic advisor of the UNIFI team. Benedetto is full professor of Robotics at the Dept. of Energy Engineering, University of Florence and has forgotten how many bills he had to pay for the construction of Turtle vehicle. A few centuries ago Benedetto used to be a (surface) Finn sailor and now is glad to be the advisor of a wonderful team in the framework of an exciting (underwater) competition.

Item	Cost in euro
Mechanical components (Al profiles, cable glandes, small parts, etc.)	606
Hummel watertight cases	322
Electrical components (Arduino board, cables, relay, connectors, etc.)	206
Magnets	120
4 ROXXY motors + 4 ROXXY regulators	500
4 LiPo batteries	443
1 LiPo battery charger	49
1 Imagenex 852 Echo Sounder	1425
1 Xsens MTi	1990
1 Depth sensor MS5535-CM Measurement Specialties Inc.	30
1 Webcam SPZ5000/00 - PHILIPS	45
1 Micro Maestro 6 Channel USB Servo Controller	19
1 Ethernet connection	68
1 WiFi receiver PICOstation	120
Total	5945 euro

Table 1: Total expenditure for Turtle vehicle

Risk for the vehicle	Prevention
Damages during transport	Most parts made of plastic. The damaged parts would be printed again.
Collisions with other AUVs	Turtle speed is very low, the vehicle cannot damage itself or the other ones.
Software failures	Time out software tasks. Thrusters will be switched off.
Vehicle recovery	Turtle will surface if the vertical thrusters switch off.
	Easy lifting thanks to vehicle low weight.
Risk for the people	**Prevention**
Injuries due to thrusters	Propellers made in plastic. Propeller casing secures the thrusters.
Injuries manually moving the AUV	Most parts made of plastic. No dangerous parts exposed.
Injuries due to lifting the AUV	Ergonomic handles. Low weight. Two people can easily lift Turtle vehicle.
Electrical injuries	Low voltages and currents.

Table 2: Risk assessment for Turtle vehicle