M.A.T.E. ROV
COMPETITION 2018
Explorer Class

POLITECNICO
DI TORINO

ITALY, TURIN

POLITOCEAN

TECNICAL REPORT

NEREO

This is our first M.A.T.E. ROV competition

Supervised by Dr. Claudio Sansoè

MEMBERS	TEAM ROLE	ACADEMIC YEAR
Sciortino Dario	CEO and CFO	5th Academic Year
Balvis Massimo	Mechanical Chief Engineer	3rd Academic Year
Bonicelli Francesco	Software Chief Engineer	5th Academic Year
Ruggiu Riccardo	Electronic Chief Engineer	4th Academic Year
Sivieri Flavio	Pilot and Fluid Dynamic Supervisor	4th Academic Year
Gallotti Davide	Software Engineer	4th Academic Year
Martellone Lorenzo	Software Engineer	1st Academic Year
Stagi Roberto	Software Engineer	3rd Academic Year
Alasio Matteo	Electronic Engineer	3rd Academic Year
Scantamburlo Damiano	Electronic Engineer	5th Academic Year
Silvestro Simone	R&D Responsible and Electronic Engineer	5th Academic Year
Calasso Giuseppe	Mechanical Engineer	3rd Academic Year
Caponio Eleonora	Mechanical Engineer	5th Academic Year
Galati Hilde	Mechanical Engineer	5th Academic Year
Musso Emanuele	Mechanical Engineer	2nd Academic Year
Brahimaj Detjon	Mechatronic Engineer	4th Academic Year
Centorrino Vittorio	Mechatronic Engineer	5th Academic Year
Greco Tommaso	Mechatronic Engineer	5th Academic Year
Longo Antonio	Mechatronic Engineer	5th Academic Year
Salvatelli Michele	Mechatronic Engineer	4th Academic Year
Masnovi Patrick	Graphic Designer	3rd Academic Year

TABLE OF CONTENTS

I. DESIGN PRINCIPLE

I.I ABSTRACT

Team PoliTOcean was born in May 2017 as the initiative of twelve students from Politecnico di Torino to create an academic reference point for underwater robotics and communications.
Due to the specific theme of the project our priority was first of all to demonstrate our ability and determination. This has been accomplished with the development of a very early and basic prototype of an R.O.V. in just three months of work and a budget of 3000€.
The prototype has been developed with a frame of PVC pipes and T200 thrusters, with a basic electronics to just allow the movement of the six motors through a serial communication with a surface laptop and a joystick.
Thanks to the success, even if in its own little, of the first prototype the University gave its trust to the project, increasing the annual budget in order to allow the participation to the competition.
In the month of October 2017 we started a recruitment campaign, receiving a huge and unexpected interest that allowed us to welcome nine more students, counting now to a total of 21 active members.

Figure I - Rendering of the first prototype

Our design is based on modularity, and this choice came from the idea to improve our ROV for future version, instead of developing a new one from blank sheets.
For the electronics we developed three different PCBs dividing the three main electronic fields of our system (Power, Control and Arm), for the mechanical part we designed four-block chassis fully customizable thanks to the simple plastic material used that can easily be drilled in order to host new components or to change the position of the existing ones. Also the computer science part is based on modules, since it uses ROS, a raspberry compatible O.S. based on nodes, and they can easily be added or removed from the system as the ROV

Figure 2 - Render of our ROV "Nereo"

II. CORE DESIGN

2.1 MECHANIC

2.1.1 Robotics

The robotic arm of the ROV has been entirely designed and developed by the team, according to the specific requests for the different tasks and it is composed of four components: a shoulder rigidly fixed to the chassis, a forearm, a wrist and a gripper. From the analysis of the movements to be done, three degrees of freedom have been considered necessary for the arm. First of all, a rotation of the forearm about the shoulder is required to level the OBS by rotating the tees, in order to avoid moving the entire vehicle in vertical position; forearm can rotate 90 degrees down from the horizontal position and 45 degrees up, to make easier the general actions of the arm. The second degree of freedom is the 360 degrees rotation of wrist around itself, that is essential for the rotation of the tees. The last is the opening/closure of the gripper to pick objects.

The rotations are possible thanks to stepper motors, the first into the shoulder for the rotation of the forearm and the one for the revolutionary movement of the wrist, into the forearm. This kind of motor ensure a precise and reliable positioning, since there are no contact brushes. About the transmission of movement from shoulder to forearm, this is possible thanks to a worm gear chosen for three main reasons: it transmits the rotation between two perpendicular axes; it provides a very high reduction ratio, necessary to have sufficient torque; it is not a reversible system, that means that the forearm keeps its position when motor is turned off. For these motivation, the initial choice of a cardanic joint has been replaced with worm gear.

For opening and closure of the gripper, instead, a linear actuator has been chosen; it is linked to the gripper with a mechanism of leverages that transform the alternate movement of the motor in the limited rotation of the two fingers of gripper itself.

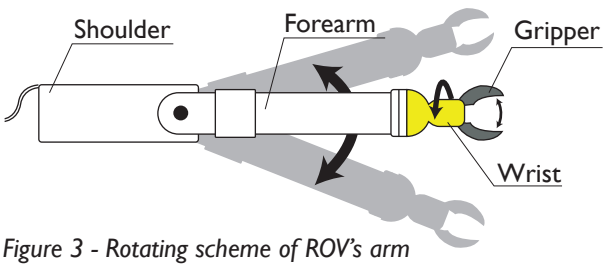


Figure 3 - Rotating scheme of ROV's arm

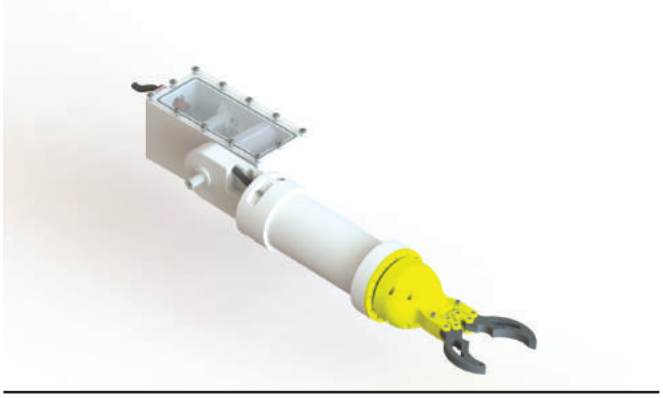


Figure 4 - Render of Nereo's arm

All the internal mechanical components are stainless steel or aluminum to reduce weight as much as possible without affecting the resistance. About the supports, next to the request of lightness there is also the one of flexibility and cheapness of their production; thus, it has been decided to use the 3D printing machine, in particular fused deposition modeling technology (FDM), available in the laboratory of the university. To compensate the inaccuracy of this technology, it has been principally chosen self aligning ball bearing to support shafts. The material used is ABS that presents a good mechanical resistance.

Also the gripper has been produced in ABS with the same technology. The design of the gripper has been studied according to the shape of the objects to catch and, to increase the friction between gripper and object, the fingers have been covered by a rubber layer. Moreover, the system of the gripper is modular, so the fingers are removable and and easily replaceable if need.

On the other side, the most important aspect of external cases is obviously the impermeability of material. Since this characteristic is not typical of FDM process, the Stereolithography technology (SLA) has been preferred; this is an extremely precise 3D printing process based on the photopolymerization, so that the material of the cases is a kind of resin, with performances similar to ABS.

Both supports, gripper and cases have been produced directly from team members, in first person, in the laboratory of the university; so that the team had the possibility to completely follow robotic arm from the design, to the production and, at the end, the assembly.

2.1.2 Chassis

The design of the chassis has started with a critical analysis of our previous ROV design. In our first ROV we used a tubular chassis, made by PVC pipes, that is very economic, easy to find on the market, but inaccurate from the point of view of the production and the mounting and not flexible to future improvements or to install other accessories not defined during the design phase.
The previous considerations and the necessity of a more competitive and versatile ROV have forced us to search another method of production.
After a feasibility study in which we have considered the costs and the production times, and after a research of a potential producers in our area, we have opted for a chassis made with plates of plastic materials, processed by a CNC milling machine.
We have chosen the most fitting material after examining the available ones from our suppliers, that is PVC, Nylon and HDPE.
All materials are easily processable by machine, PVC and Nylon have a higher tensile strength when compared to the HDPE, but we have chosen the latter, firstly for the lower percentage of water absorption, which is a very important feature in our field and, secondly, for the lower price.
The chassis is composed of four main components: one panel on the top, one panel on the bottom and two lateral panels (Figure 5).
The two lateral panels are connected to the top and the bottom panels using threaded inserts that allow to have a safer connection between parts. The threaded inserts need very large holes to be correctly installed, therefore it was necessary to use sheets with a thickness of 15 mm to maintain structural integrity.
The current shape of the plates has been chosen for several reasons: to minimize disturbances in the flow of thrusters, to reduce weight and to respect the dimensional constraints imposed by the manual (the length of the assembled chassis is smaller than 64 cm).
Other reasons concern modularity, as this chassis configuration allows to install additional accessories in an easy way.

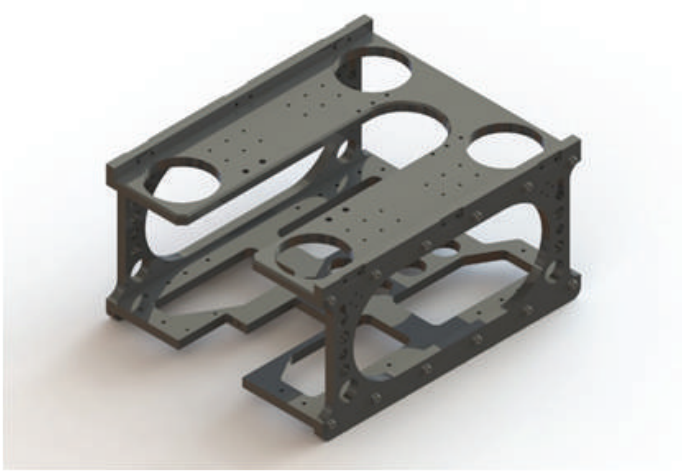


Figure 5 - Isometric view of the chassis

2.1.3 Electronic Enclosure

As for the electronics container, we decided to purchase an acrylic tube (PMMA), transparent (in order to directly see the PCBs and the control LED), 6 mm thickness, more resilient than a common glass and cheap enough. On one side the dome end cup guarantee a wide vision to the camera and the sealing is obtained thanks to O-rings on the caps. We decide not to directly manufacture the tube since any possible flaw would have been risky for the electronic components. In order to simplify the operations on the PCBs the electronics container can slide on aluminium support guides (with chamfers to avoid sharp edges) which were milled in the laboratory and it can rotate from the horizontal to the vertical position. We decided to use threaded bars to hold PCB supports for several reasons: optimising room using as much space as possible, diffusing heat and avoiding the direct contact of electronic components.

2.1.4 External accessories

Our ROV is equipped with two cameras, the first installed inside the electronics enclosure, the second placed on the arm, that allows to have a better view of the surrounding environment and to help the pilot in the execution of the tasks.
The camera installed inside the electronic enclosure doesn't need a waterproof support because the environment in which it is mounted is obviously already waterproof.
It's not the same for the arm's camera (Figure 2), for this last one we have designed an external waterproof enclosure, composed of a support manufactured by SLA 3D printer and a transparent plexiglass plate properly processed and installed in the forward part of the support, the enclosure is sealed by an O-ring seal mounted between the support and the plexiglass plate.

We have used the same technology of 3D printing to make the support of the ultrasonic transmitter (Figure 3), in this case, in addition to the O-ring seal used in the back of the support, to complete the sealing, we have put epoxy resin between the ultrasonic sensor and the support.

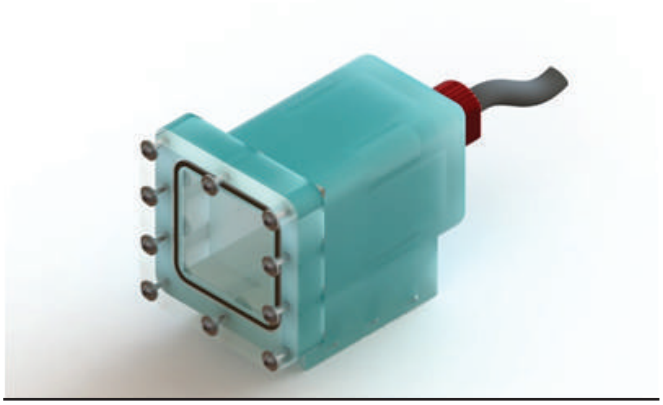


Figure 6 – Arm's camera enclosure

Figure 7 – Ultrasonic Trasmitter

2.1.5 Thrusters

Our ROV relies on 8 BlueRobotics T200 thrusters with ESC, in a 4 vertical-4 horizontal with 45 degrees angle configuration.
This choice has been made for 4 main resons : power,reliability, controll and future improvement.
Nereo, with a 12 V power supply, is able to generate up to 11.5 kgf thrust,considering turbulance loss due to flux interferences, in a forward motion and 20 kgf in a upward motion at full power.
In addition to the known T200's reliability, in case of a thruster's malfunction the system is redundant enough to finish the mission.
With 4 vertical propellers and our stabilization software, Nereo can maintain depth and trim, also when moving a load with the robotic arm.
In the future our plans involve switching the 12 V power supply with a 16 V one and adapting the electronics, with a predicted improvement of the overhall thrust by 40 %.

2.1.6 Buoyancy

Our ROV's buoyancy is provided by a 3D printed ABS casing filled with polyurethane foam.
3D printing allowed our team to design hydrodinamic shapes and manufacture the final product in an inexpensive and rapid way.
The casing is connected to the chassis by 3 hinges for side, providing an easy access to the electronic's casing by rotating out of the way.

2.2 Electronic

2.2.1 Control pcb

Our control PCB fits a Raspberry Pi 3 model B, an ATmega328P and a bunch of IC and MOS.
These PCB was built with the aim to have an hardware platform to manage all and only the logic-control signals needed to control both power and actuation (delegated to the respective PCBs), so that in the future we can change drivers, power circuits and similar without the necessity to change the control too and vice versa.
The Raspberry is responsible for live communications, video stream and arm control whereas the ATmega is only in charge of the thrusters behaviour and consequentially the ROV's motion.
The ATmega has its own I²C network connecting current sensors of the propellers, IMU and barometer (that are the three components used for ROV handling in pool space), whereas the Raspberry has another I2C network for arm signals.
The Raspberry is connected to the ATmega with SPI, both for programming and communication purpose.
The Raspberry GPIO ports and USBs were desoldered and changed with 2.54mm header pin on the bottom side to be used as a shell, inserted on the female headers on the PCB to reduce wiring, use less space and improve the hardness of the PCB. The ATmega is on the other side of the PCB with its proper socket to be accessed in testing and prototyping phase.

Since the high logic level of the raspberry works at 3.3V while the ATmega and the PCA9685 (PWM controller with I²C interface) works at 5V, a bunch of level converter are used on this board to keep the right logic level in SPI and I²C interconnections.
Signals connection were developed paying attention to their own limits and necessity. We tried to avoid the usage of any angle higher than 60° in the routing topology in order to reduce the reflection factor in the propagation signals. Moreover, in particular, USB and I2C links support an higher frequency workload (≥MHz) and so all the parallel connections of the same group should be almost of the same length.
A PCA9685 is used on our board since lot of simple control signals works in PWM, such the servo one for the tilt of our main camera and underwater lighting LEDs, and this component is capable of 8 PWM outputs controlled by just one simple I2C input signal so that we have the possibility to extend the PWM capability of our control PCB (since all the possible ports of both ATmega and Raspberry are fully used).
As said, ROV housing hosts 3 different PCB and the control one clearly has many connections with the other two. Moreover, the disposal space is not too big and so we came up with the necessity to reduce to the minimum all the wiring space. So developing our boards we paid great attention to the distribution of the connectors over the PCB placing them strategically near to their usage goal reducing consequentially the interconnection wires lengths.

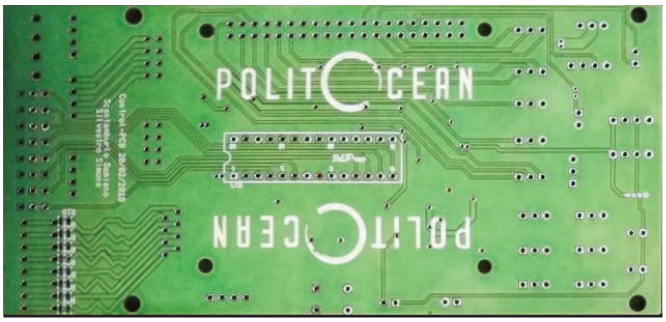


Figure 8 - Control PCB

2.2.2 Power PCB

The second PCB that we developed is entirely dedicated to power management. This allowed us both to optimize the space inside the housing and to isolate delicate circuits such as logical ones from possible voltage spikes that used to occur in our previous ROV prototype.
One 48V/5V DCDC converter and a 3.3V LDO voltage regulator are used to supply both the two microcontrollers and to the various peripherals and ICs.
Two 48V/12V DCDC converter are instead used in a parallel configuration, in order to supply enough current to the thrusters and to the robotic arm, without having too much heat to be dissipated on one single converter. Paying attentions to safety, an optocoupler was used to allows us to enable or disable this two converters via software, in order to quickly stop any movement in case of possible hazards.
The presence of the three main supplies are moreover indicated by three, different coloured LEDs, allowing us to immediately see if our system is powered or not, if the 12V is active and to recognize possible power supply problems.

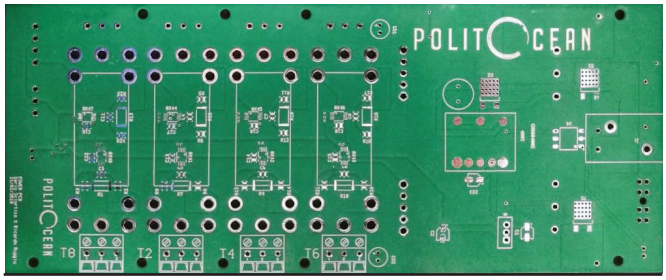


Figure 9 - Power PCB

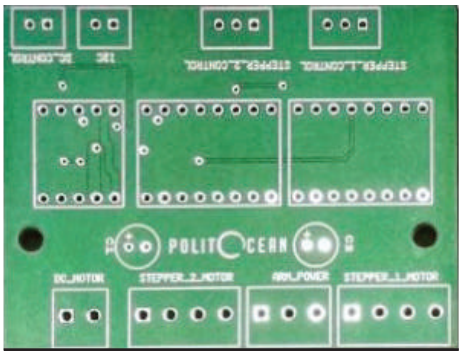


Figure 10 - Arm PCB

2.2.3 Arm PCB

The third PCB is instead dedicated to the handling of the mechanical arm.
This PCB manages the three motors placed in the arm: two steppers used for the shoulder and the wrist, and one linear actuator for open and close the claw.
This PCB as said is connected to the I²C bus of the Raspberry and also to a bunch of GPIO pins to control the drivers of the steppers, whereas the power is provided directly from the power PCB. Inside this board there are two ICs with I²C interface: one is the INA220 responsible of reading the current from the linear actuator and the other is a multiplexer that sets the resolution for the steps. Here there are also incorporated three drivers: two DRV8825 for the steppers and a MAX14870 for the actuator. Those drivers are not directly soldered on the PCB, but they are connected with the female header soldered on the PCB. This solution was adopted because the drivers can break after a long usage(due to the temperature), in this way replacing a broken driver is very easy and not stressful for the pads of the PCB.

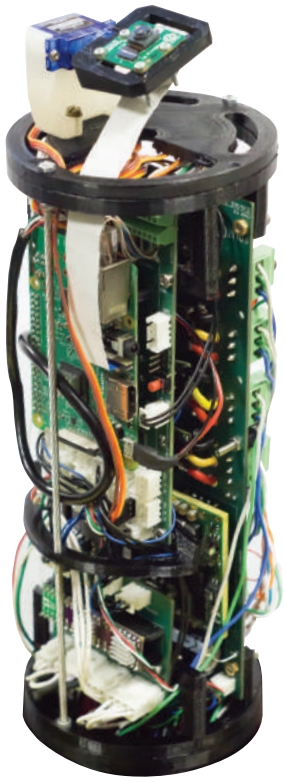


Figure 11 - PCBs

2.3 SOFTWARE

2.3.1 Core

The system for the communication of the components in the ROV is based on ROS (Robotics Operating System). For this reason we decided to use a Raspberry Pi hardware system in which we installed all needed softwares that we use. There are two Raspberry boards: the main one is in the ROV while the other is mounted on the ground station and the two are connected together through a network cable. A first solution was to use only an ATMega as micro-controller because we thought that having to learn and implement ROS was too much of an effort. The main reasons that brought us to choose ROS is its modularity: indeed it was a great advantage for the development process and the system gains a significant scalability. Furthermore, this modularity allows us to eventually include external developed nodes, in fact the camera nodes are taken from the ROS libraries (1),while the other are custom.Finally we decide to implement our scripts in Python instead of C++, because the last one needs to be compiled affecting the prototyping phase. The following picture represents a simplified architecture diagram of our ROS platform (figure 12).

The terminal node is the GUI, executed on the ground station, useful for the visualization of the data written in the ROS topics. It is developed with the PyQT library (2). It also contains a console text module to communicate with ROS, to display error messages and to send control signals. Regarding the electric motors we decided to use an ATmega328P since they are driven by a PWM signal, requiring a quite stable frequency not obtainable from the Raspberry Pi. The ATmega is also responsible for the self-stabilization control algorithm: the navigation input data are taken from the joystick and the Raspberry is in charge of sending them to the ATmega. This is due to the fact that the input libraries are written in Python.

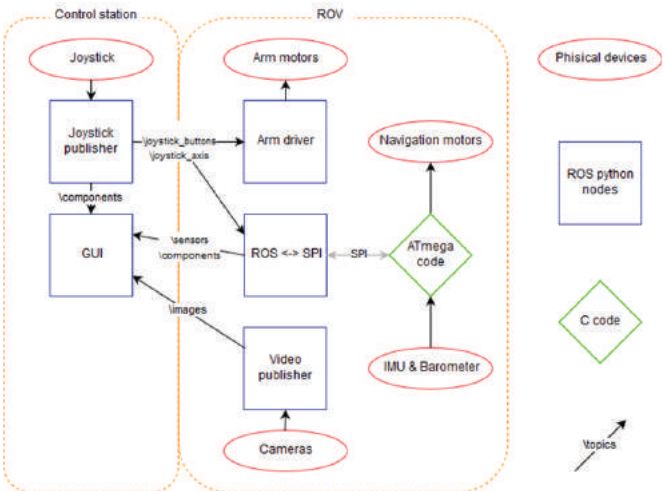


Figure 13 - Flow chart

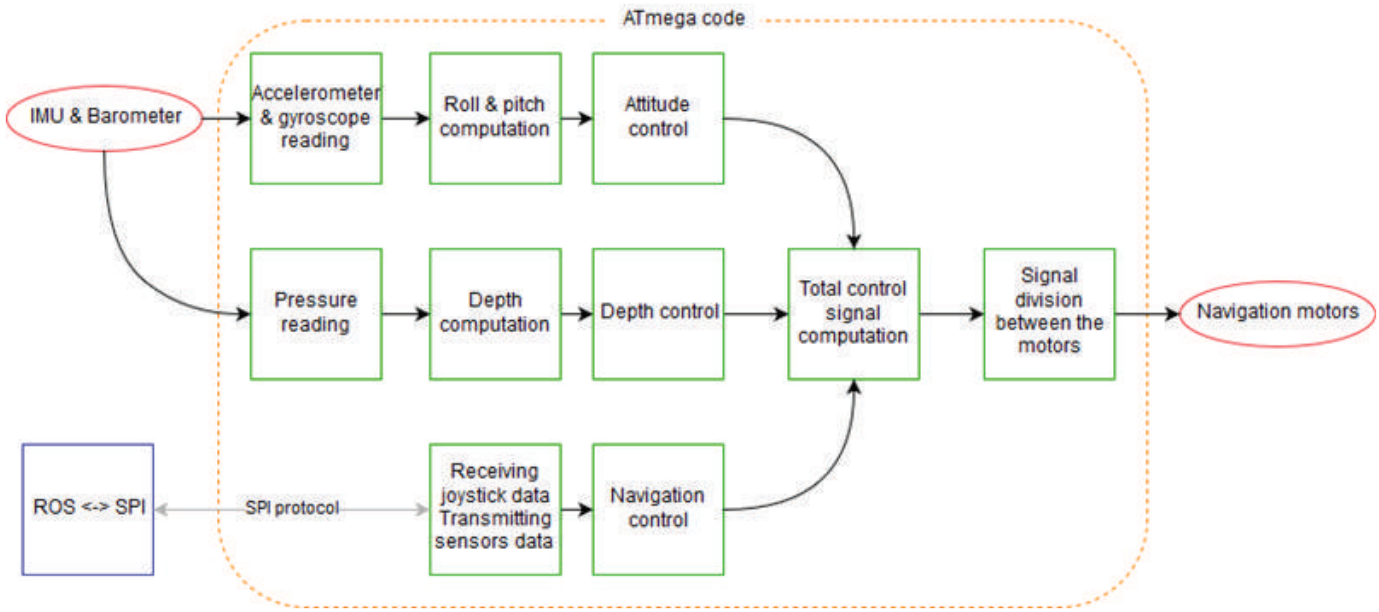


Figure 12 - ATmega code flow chart

2.3.2 Control algorithm

The algorithm is written to implement a purely proportional controller. Initially it was designed as a PID controller, but we discovered that for our purposes it was enough simply a proportional one. A software saturation is also implemented to take into account the physical limitation of the motors and to reduce the output current. Through the definition of a reference point and an error function the proportionality constant is found experimentally: in this way the attitude of the ROV is correctly stabilized when the measured values are too different from the reference ones.
To obtain the angle measurement we initially started using an atan2 function: this resulted in inaccurate results since it is an approximation unable to take into account all the necessary computation factors. For this reason, we decided to compute them through the inversion of the rotation matrix. In the early stages we tried to do the complete inversion during every control loop: this was found to be too heavy both computationally and memory-wise. In the end, we decided to directly compute the trigonometric quantities arising from the matrix product and use them to derive the angles.

(1) http://wiki.ros.org/video_stream_opencv
(2) https://wiki.python.org/moin/PyQt

III. CONTROL BOX

The core of the control box is a Raspberry Pi 3B, which is part of the ROS network and runs the GUI, allowing us to send commands to the ROV and obtain data/video from it. This Raspberry is mounted into a waterproof case, modified with two 3D printed frames: one to support the LCD screen, on the upper part of the case, and another on the bottom to house the Raspberry, the LCD converter, two DC power suppliers(12V and 5V) and a cooling fan. The ports of the Raspberry, four USBs and aEthernet, are accessible directly, without any further connectors. The Joystick, mouse and keyboard are connected to the USBs while the ROV is connected to the Ethernet. An additional connection is made from the GPIOs of the Raspberry through an extra ethernet connector to power and import data from the OBS that, with a single ethernet cable, is supplied from the 12V DC power and sends the data through the serial line of the Raspberry.

Figure 15 -

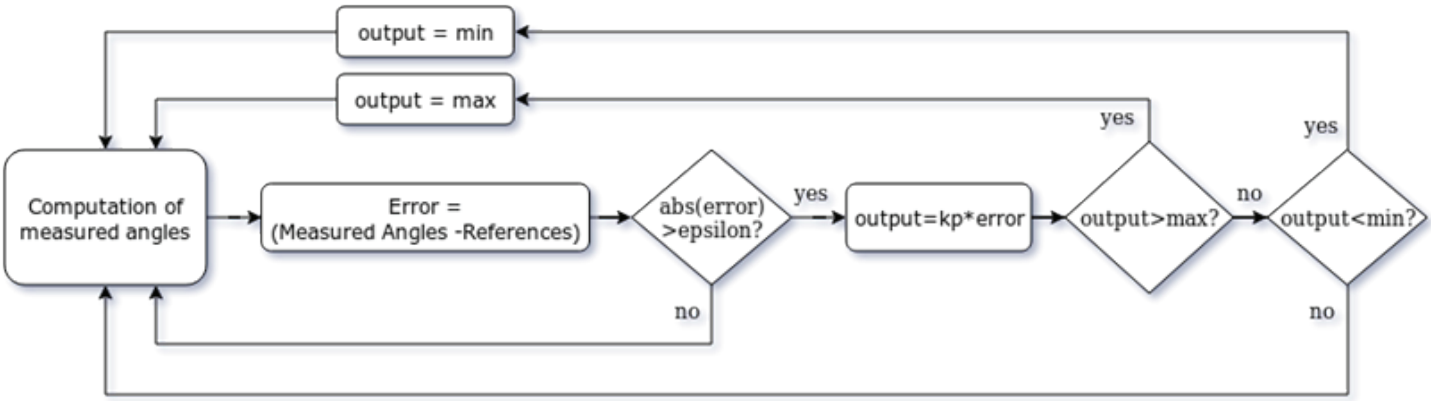


Figure 14 - Flow chart

IV. TROUBLESHOOTING AND TESTING

4.1 SEALS

One of the most important topic during the design of the arm was the choice of the way to seal the shafts and the enclosures. Traditional seals were initially used but during the tests we discovered several problems, such as the corrosion of the metallic element of the seals that were not designed to work in water and the increase of friction due to pressure on the lip of the seals, blocking the rotary motion. The test was conducted with an instrument realized in our laboratory, composed by a cylinder and a piston who simulated water pressure on the seals. For this reason we adopted custom seals made from a local factory that in according to our needs have been built in NBR material for a roto-translating motion and with resistance up to a pressure of 10 bar with a low friction.

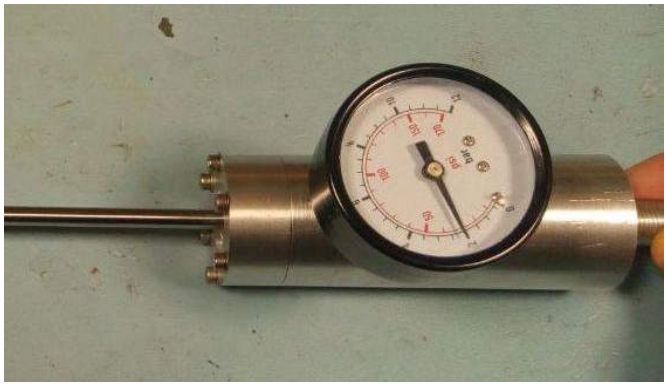


Figure 16 - Our hyperbaric chamber for seals tests

4.2 ELECTRONIC HANDLING IMPROVEMENT – EASIER ACCESS IMPROVEMENT

Since in the previous ROV one of the most frequent problem concerned the direct operations on the PCBs, as it was so hard to directly work in the small area inside, we used a sliding system for the tube so that it can rotate from horizontal to vertical position.
As for the location of the IMU we placed it in the center of gravity, whose position has been calculated on Solidworks, joining the sensor to the whole system through a light polymeric support.

4.3 VIDEO

The real-time visualization on Ground Station is not simple to obtain. The problem has born because the ARM CPU on the Raspberry is not good enough to elaborate the frames coming from the cameras. Initially we worked at high resolution which we later found to be excessive for our real need. We resized the frames with an OpenCV (1) library using an interpolation algorithm: it wasn't enough because resizing time was higher than the one needed to display the frames without delay. In the end we decided to fix the camera resolution at a low value (640x480) without resizing.

4.4 IMU CALIBRATION

The calibration of the gyroscope is experimentally computed. Initially we 3D-printed a structure able to have a known angular velocity through a little DC motor. We had some logistic problems with cable management that forbid us to use this method. Finally we made the calibration through recursive measurements of known angles: this gave us some values close to what we expected and with some little adjustments we obtained the needed constants.

4.5 JOYSTICK LIBRARY

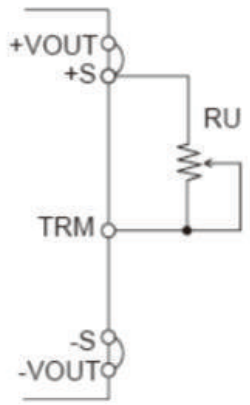
Initially the reading of the joystick input was done with an external library, but it was found to be incomplete (some buttons were not properly recognized). For this reason, we decided to substitute it with a custom one.

4.6 PCBs

One problem that we had to fight is related with powering Raspberry.
During some phases in which the Raspberry on board had to manage large amount of work such as the video stream, we noticed how its red power led used to turn off for some milliseconds. After some research, we discovered that this behavior is related to insufficient voltage at its main power input (below 4.6V, in particular). We identified the problem making some test using an external power supply for the Raspberry and trying to find out when the problem arise and which peripheral provokes it. As a result we observed that, with the external power supply the Raspberry power red light did not blink anymore also with the maximum workload possible. Consequentially we found out that adding a potentiometer between two dedicated (not used before) pins of the 48V/5V DCDC converter, we managed to adjust the voltage in order to read at least 5.0V at the main supply of the Raspberry.

Figure 17 - Cross check of the design

Another aspect that we have to deal with was signal integrity.
One of our priority is to reduce as much as possible the length of the internal cables in order to ensure certain clearances between them. One of our previous solutions were manually solder cables but they were too fragile and sometimes too loose to guarantee secure and stable connections between PCBs.
That's why we decided to adopt JST crimped cables, allowing us to both have some solid even if small connectors and to create some custom-length wire without having to solder anything.
The third main aspect we took into account is the connection between the Control station and the ROV through an ethernet cable, wich is responsible of all the communication between the two.
Last but not least of all the troubles there are stepper and dc-motor drivers, which are very delicate components and need a fine tuning in order to work properly without breaking, they need particular attention on the right positioning of the trimmer that control the current provided. We tested all the drivers with their linked servo motor before placing them onboard, anyway, after placing the component on the arm PCB they worked with slightly different behavior due to different connections (on and off PCBs) and to the environment wich is more complex in pool tests using all the demo props.



$$RU = \frac{5.11 \times V_{OR} \times (1+\Delta)}{1.225 \times \Delta} - \frac{5.11}{\Delta} - 10.22\ [k\Omega]$$

$$\Delta = \frac{V_{OU} - V_{OR}}{V_{OR}}$$

V_{OR} : Rated output voltage [V]

V_{OU} : Output voltage needed to set up [V]

Figure 18 - Schematic modification for 5V output adjustment

V. PERSONAL CHALLENGES

5.1 MECHANICAL

Since we are multidisciplinary team, our cultural limits have been pushed to achieve proper technical communication and collaboration between all the members.
We also had to merge together both academic and teamwork commitment with all the related planning problems, always aiming to meet deadlines.
We improved our purchasing management abilities, especially in terms of relationship with suppliers and academical reference; moreover, we worked towards optimizing budget, shipping time, manufacturing processes and sponsor commitment.
Concerning the materials, we studied and compared several characteristics to use the most suitable products in terms of cost, mechanical and thermal properties and availability. Since we manufactured lots of components by additive, we had to deal with different features of both materials (ex. glass transition and melting temperature, resilience, waterproofing...) and printers (ex. pre - and after printing treatments for resin, infill percentage, plate and extruder regulation...). As for the chassis we studied and compared several solutions before choosing HDPE for its mechanical properties (besides moderate price, and easy accessibility).
We learnt to use Solidworks software not only for CAD drawings, but also for motion and flow simulations (in order to optimise thruster's location) and for calculating mass properties.

Figure 19 - MJP 3D printed arm port

5.2 ELECTRICAL

First of all we learned well how to use PCB CAD for schematic and layout, managing our custom database and sometimes creating footprints from scratch.
Since we use SMD components (from basic 0603 resistors/capacitor to some IC like INA220 or PCA9685) we learnt to use solder paste, pick and place and infrared IC heater. We also had lot of through hole components so also the common soldering station has been used, always coupled with aspirator to aspire all the soldering smoke.
Another big challenge has been communication with all the protocol used on our system (SPI, I^2C, USB, serial, Ethernet) with attention both to hardware and software configurations, to ensure a robust connection trough all the logics.

Figure 20 - Learning the Pick&Place machine

5.3 SOFTWARE

Since this team was born only one year ago, the hardest difficulty concerned acquiring a huge cultural baggage of technical knowledge, needed to start building the ROV software platform. The main goal reached by everyone is the acquisition of a good knowledge of ROS: this includes both the ROS system and the Python programming language, although no one had used it before because it is not part of our academic background.
To implement the video-stream and the computer-vision task it was necessary to learn OpenCV library and some basic multimedia elaboration, while we had to learn the PyQT library to code the GUI (Graphic User Interface).

VI. SAFETY

6.1 THE 12V ISSUE

From an electrical point of view, one of our main concerns was the possible uncontrolled motion of motors such as thrusters and arm parts during code debug phases. The simple solution of the optocoupler described above allowed us to safely modify the code, enabling the 12V output only after ensuring that no one was handling the ROV in any way.

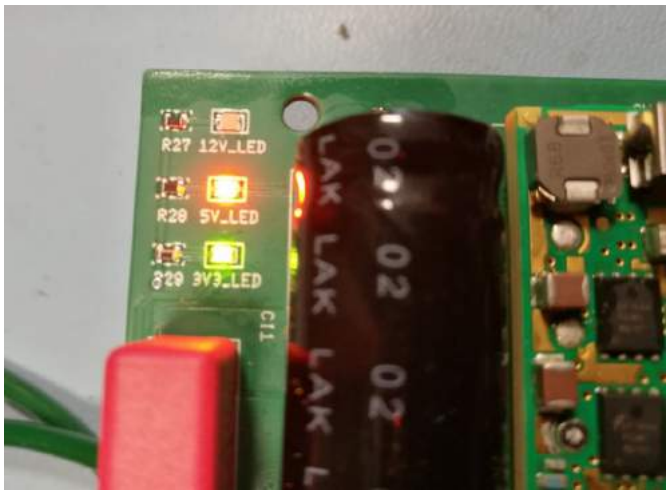


Figure 22 - 12V feedback

6.2 CONTROL STATION

The control station has been developed with particular attention to safety, the waterproof case has not been touched and drilled in order to not compromise its waterproofness, the 220/110 plug has its fuse, we used stickers to clearly sign where the high voltages are, the cable color convention is clearly specified by the IEC sticker (EU standard), and there is a transparent plexiglass windows to always have a clear view of the inside of the case.

Figure 21 - Team member at work in laboratory

6.3 ADDITIVE MANUFACTURING

One of the main risk concerning the 3D printing consist of inhaling some substances, such as styrene (from ABS) and to avoid any danger we always worked on a well-ventilated environment. Moreover, the safety lock device of the printers prevented the direct contact with the hot extruder in action and in order to touch the hot plate after printing we used appropriate gloves and trimmer to take out the products.

6.4 SAFETY AT THE WORKSHOP

Since safety concerns are of paramount importance, we paid attention to prevent any kind of risk, especially at the mechanic workshop: protective glass and gloves, besides clothing as protective suit and shoes, were always used by the operators. All the machineries (such as milling machine, grinder, saw...) were only use by competent staff, some team mates who followed a brief training course.

VII. BUDGET AND TIMELINE

7.1 BUDGET

As our first year of competition we didn't actually had anything to be recycled from previous years. The only things we could have recycled were the six T200 motors and the acrylic tube for electronic housing, but we decided that it was worth to maintain the old ROV prototype unbroken in order to have it ready for possible tests or demonstrations.

Moreover, being a newborn team, we didn't have any kind of storage components so we also had to procure a certain number of useful ones to be stored for possible uses. This is particularly referring to passive electronic components and ICs, screw and nuts and other must-have items.

Purchases have been very challenging since being part of a public administration we're not allowed to directly buy items, but rather to issue a public call specifying the characteristics of the requested components. This is of course a long process so we always had to think ahead and make the most of every order.

As shown in the tables the raw cost of the ROV and its Control Box has been calculated to be 8.513,37 €.

Apart from the sponsor incomes that has been received in the form of items, our main and only investor is the Politecnico di Torino that for this year funded as with a total of 30.000€.

The remaining, not mentioned, amount of money has been used for purchases and investments that are not strictly related to the competition, such as: testing equipment, t-shirts, advertising materials and refunds of travel expenses for local ROV companies visits.

Electronics	Cost
Passive components	219,08 €
ICs and sensors	143,32 €
PCB printing services	450,52 €
Boards & prototyping material	215,96 €
Power converters	296,68 €
Connectors	242,01 €
Cables	190,21 €
Cameras	227,87 €
TOTAL	**1.985,64 €**

Thrusters & motors	Cost
T200	1.893,44 €
Steppers and motors	132,25 €
TOTAL	**2.025,69 €**

3D printing & prototyping	Cost
Tough resin for SLA 3D printer	582,00 €
MJP printing service	204,49 €
Total	**786,49 €**

Mechanical	Cost
Electronic housing set	372,10 €
Chassis	500,69 €
Screws and nuts and gears	595,54 €
Shafts	280,08 €
Bearings and seals	182,54 €
Raw materials	323,07 €
TOTAL	**2.254,01 €**

Control station	Cost
Joystick and peripherals	269,78 €
Electronic components	308,22 €
Plastic box	69,54 €
TOTAL	**647,54 €**

Tools	Cost
3M Epoxy	325,74 €
Tools	224,20 €
Glues and adhesives	62,35 €
Safety equipment	18,85 €
TOTAL	**631,14 €**

Sponsors & Donations	Cost	Donator	Notes
2 Kgs of PLA and ABS filament	64,00 €	Filoalfa	20 Kgs have actually been donated
48V power supply	500,00 €	Mouser	
Costom rotarty shaft seals	250,00 €	SKF	
TOTAL	**814,00 €**		

Total ROV expenses	Cost
ROV expenses (including donations)	8.513,37 €
Tools expenses	631,14 €
Travel expenses	11.250,44 €
TOTAL	**20.394,95 €**

Travel for 9 members	Cost
Plane tickets	8.116,11 €
Registration Fees	260,00 €
B&B booking	2.874,33 €
TOTAL	**11.250,44 €**

7.2 TIMELINE

It has been very challenging to meet deadlines while developing the systems and training the new members to the ROV and in general to the underwater robotics word, but splitting the work between the three well-structured sub-teams we managed to do it.
Our original target was to complete the design phases by the end of December, exploiting the external production times to have a break and study for midterms.
Unfortunately, the arm and PCBs design took more than expected but we luckily managed to work even during the "winter break" to let the production start with an acceptable delay.

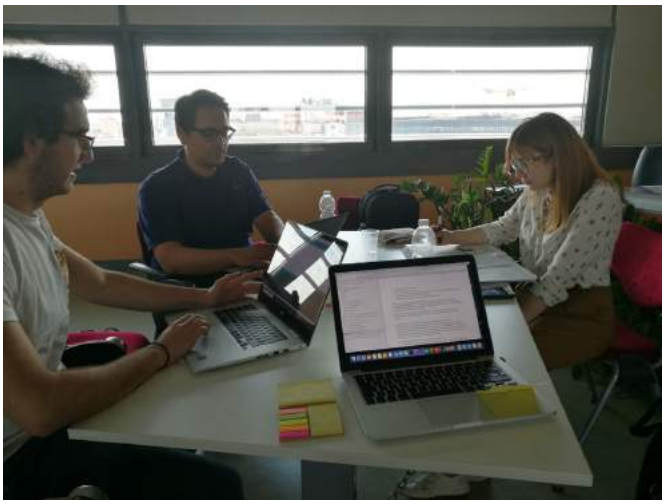
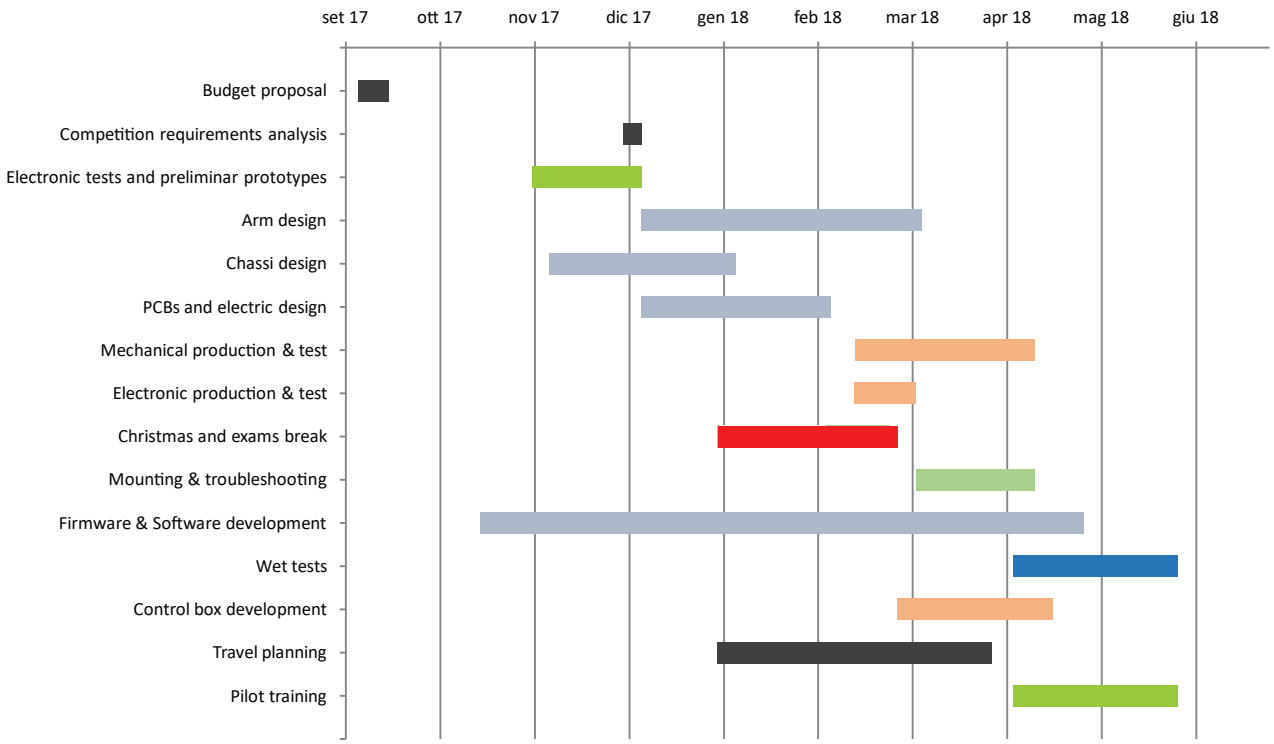


Figure 23 - Meeting



VIII. FUTURE IMPROVEMENTS

While testing our solutions, we had compromises in order to achieve cost and first of all time constraints, but we're looking forward to improve some aspects in the next ROV generation.

8.1 ELECTRONIC

At the moment, the ROV's arm has not any feedback in terms of position, so the pilot doesn't actually know how the arm is oriented unless he understands it through the video images. Two possible solutions have been discussed: the addition of encoders would allow us to know the position at any time with the best precision, whereas the addition of two endstops would instead avoid the pilot to reach mechanical limits and to know the position even if in a relative, less precise, way. Another improvement is related to autostabilization and autoquote, at the moment we can level the ROV to a known depth and set it parallel to the horizontal plane using vertical motors: next step will be to use an IMU with the built-in magnetometer to stabilize the ROV also in its orientation referring to the north, and so block it in a precise position with respect with all the possible axes. Referring to the tether, right now we use 2 separate cables for power and ethernet communication, in next generation ROV we want to test a single cable including both power and ethernet for robustness and for easy cable management, or maybe try other type of connections such as optic fiber.

Figure 24 - Learning from a local ROV company

8.2 MECHANICAL

The first big mechanical upgrade that we want to try is the reduction of the total weight of the rov, using lighter material for the chassis, and improve all mechanical components using topological optimization.
Another improvement can be represented by a little wider housing for the electronic, useful to better manage the internal space for heat dissipation (maybe use an aluminum housing to dissipate the heat with the pool water).
In order to cut down the total cost of the ROV and to set our self a big goal, we're thinking that in the future we may develop our own thrusters, since the T200 by BlueRobotics that we use are quite expensive and we even eight of them.

8.3 MECHATRONICS

For the next year we want to improve our controller from the current Proportional to a PID. An adaptive control strategy, like MPC, combined with an estimation process of the ROV model, would be ideal. Furthermore, we want to add a horizontal stabilization in order to give to the driver the possibility to maintain a certain position. About the arm we want to move it in a position in order to increase the preciseness of the movement and also to give more control of this part to the driver. Regarding the clamp and so the actuator we want to control it by current: doing this we will be able to choose the couple applied by the clamp obtaining a more robust component.
For what concern the testing part, we have in plan to build a SW model of the ROV, including all the command we will use in the physical one, in order to be able to test our scripts and then analyze the behaviour of our ROV.

IX. ACKNOWLEDGMENTS

Our first important thanks goes to our university "Politecnico di Torino" that gave us the chance to partecipate and work on this project giving us the means and the financial resources needed.

Team PoliTOcean would like to thanks its sponsors, that gave not only an economical help but a great encouragement and a good mind to go on:

Mouser electronics, that kindly donated the 48V power supply and the two DCDCs used in the previous ROV.
Bluerobotics that sent us free a T200 thruster during our early stage, in order to rapidly get the hands on something .
SeaOutpost, the local Bluerobotics dealer that always offered us free and super-fast shipments, as well as precious suggestions on the ROV building and mantainance.
Life Comunicazione, for its kindness on creating us a web site with no expenses.
Filoalfa, an italian 3D printer filament producer that provided us a massive amount of PLA and ABS filament to create our prototypes and not only.

We also think that we couldn't have made it without the logistic support offered by the local acquatic society *Torino Sub*, the community pool and in particular the *Vigone* pool that provided us an important and useful access to acquatic areas.
Fast test have instead been conducted in our small and tight pool that has been kindly hosted from the Environmental and Territory department of our University.

Clearly we would like to thanks the MATE Center which organised the competition giving us a goal to aim and the possibilities to grow as a community..

But the biggest thanks go to our professor and mentor Prof. C. Sansoè, that is a professional scuba diver and believed in our project since the first moment.
He always left us free in terms of technical aspects, trusting us, but at the same time being always available and suggesting solutions in case of troubles.
More than anything, with an enviable patience, he helped us to fill that gap that was originally present between us students and the University as a public administration, teaching us how to deal with the bureaucratic affairs.