ASUQTR: SUB 2.0

*Note: Sub-titles are not captured in Xplore and should not be used

Bastien Côté
Department of Electrical and Computer Engineering
Université du Québec
Trois-Rivieres, Canada
Bastien.cote@uqtr.ca

Gabriel Tremblay
Department of Electrical and Computer Engineering
Université du Québec
Trois-Rivieres, Canada
Gabriel.tremblay@uqtr.ca

Joel Bisson
Departement of Mechanical Engineering
Université du Québec
Trois-Rivieres, Canada
Joel.Bisson@uqtr.ca

Marc-andre Morrissette-Soucy
Department of Electrical and Computer Engineering
Université du Québec
Trois-Rivieres, Canada
Marc-Andre.Morrissette-Soucy@uqtr.ca

Mathieu Labranche
Department of Electrical and Computer Engineering
Université du Québec
Trois-Rivieres, Canada
mathieu.labranche@uqtr.ca

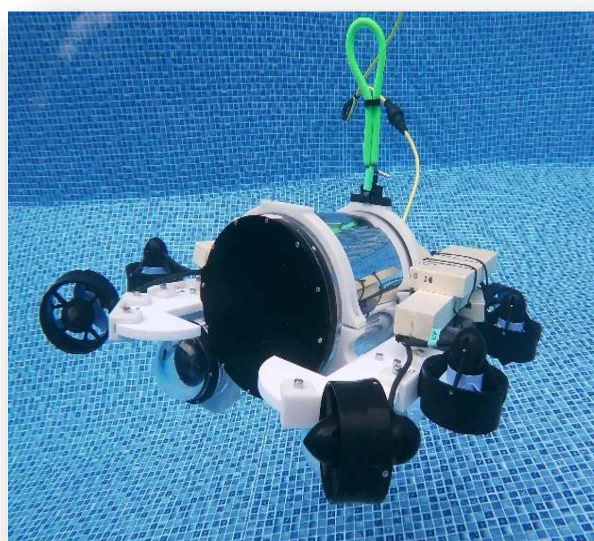
***Abstract*— This paper presents Autonomous Submarine Université du Québec à Trois-Rivières (ASUQTR) team's competition strategy and design decisions for its second participation in the Robosub event. It will discuss the progress achieved within the hardware, software, testing and management aspects of the project. These improvements are the core of the team's creative and engineering process. Therefore, they will be described to showcase how they participate in applying ASUQTR's competition strategy. Being a small team with limited budget, ASUQTR aims to design most of their hardware/software with optimized costs and simple yet effective concepts. Thus, every subsystem is implemented to score competition points reliably and modestly while avoiding complexity in development. An example of such an implementation is the linear quadratic regulator (LQR) control algorithm for navigation, which allows simple and fast configuration. To achieve further complexity, each architectural submodule is carefully designed with modularity in mind. This allows to easily increment/decrement the AUV without breaking the system. This modularity is mandatory for efficient management within the team and the project scope (scalability, members turnover, educational context, etc.).**

Keywords— AUV, ASUQTR, RoboSub Competition

I. COMPETITION STRATEGY

This year (2021) is the third year of the ASUQTR students club. The current AUV prototype is the third version. It is the first version that includes a feature allowing to achieve each kind of challenge in the Robosub competition. This year, the team's current strategy is divided into 3 components: reliability, simplicity, and diversity.

Reliability, in this context, means that a solid ground for basic objectives shall be achieved. For example, gate and navigation are considered basic objectives. This means that points associated to theses missions shall be confidently attainable.



Hence, a lot of testing time is reserved for laying out solid grounds for the hardware (electronics/hull contents) to avoid problems while in higher layers of testing, such as software components and autonomous behaviors. The second major portion of testing time is reserved for the control system configuration. This will result in minimal bugs coming from hardware and control system. The remaining testing time, even if minimal, can then be used efficiently for more complex objectives.

The simplicity component is expressed through the team's design decisions. Each subsystem is conceptualized and developed in its own scope to enable fast and clear testing results. This also allows quick integration with the whole system and minimal bridges between subsystems. While this seems obvious for basic AUV components, this mindset really shines when the complexity is quickly increasing, such as when designing autonomous behaviors. For example, the decision to encapsulate every Robosub missions individually improves the testing times by making the transitions easier and feedback logs clearer. This way, the team may rapidly choose to avoid a mission while trying to achieve the others that are more likely to score points. Each of these missions has linked micro-behaviors encapsulated in their own scope, which also allow them to be thoroughly tested and optimized.

The diversity in strategy is to have several functional subsystems available. This is useful improve de versatility of AUV and adapt to different challenges we face while testing in and out of competition context. Throughout the year, hardware modules are added to expand the software team's options. For example, a different kind of sensors can be exploited help complete a certain mission step easily compared to the previous implementation. As such, ASUQTR's newly acquired Dopler velocity log (DVL) sensor greatly improved the precision and management of the navigation system. Machine learning algorithms are used for object detections, but alternate camera streams also use simpler and faster algorithms to spot buoys. All the modules required for more complex challenges will be available for their corresponding objectives. The actual implementation of the mission behaviors depends on the testing time remaining before the competition. Some of those modules are the droppers, torpedo launcher and the gripper.

In summary, the competition strategy is to have everything ready and problem proofed (reasonably) on the hardware side of things for the competition tests and time slots. This way, a maximum number of basic tasks and autonomous behaviors can be implemented with robustness and repeatability, while having the possibility to complete more complex but less tested parts of the challenges.

II. DESIGN CREATIVITY

In this section, the evolution of the AUV's design since Robosub 2019 competition is presented. The justifications for those changes are also presented. Each requirement was defined and adjusted according to the Agile SCRUM development methodology.

Mechanical

Requirements:

- Modular and versatile vehicle
- 100m underwater vehicle with embed electronic and batteries.
- 6 DOF (Degrees of freedom) vehicle
- 4 cameras embed on the vehicle
- 1 gripper on the vehicle
- 1 torpedo launcher
- 1 dropper module
- 1 positioning system
- 1 acoustics module

- 1 collision detector captor.

When designing a frame, it is important to consider a modular assembly in order to manage complexity as the project progresses. Hence, the choice of a SLS (Selective Laser Sintering) 3D printing nylon frame allowed a precise printing with 316 corrosion-resistant brass inserts. Coupled with a waterproof finish and shock resistant characteristics, these design choices are solid proof of concept for a scalable main frame. Without this feature, it would not have been possible to handily add hardware modules to the vehicle.

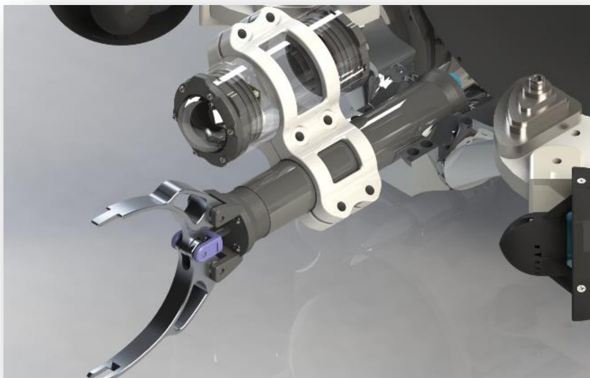
Another creative addition is the thruster support setup, which is made of two 1/2" thick acrylic plates. This configuration grants the ability to move the engines as needed and to dispose weights in the slotted holes to balance the vehicle under water. In order to always oblige to the modularity concept, the 30 x 30 aluminum profiles from ASUQTR's AUV 1.0 were ingeniously fitted underneath the AUV 2.0 to add modules over time. In the same vein, AUV 1.0 batteries were reused in two new 4" enclosures for better space management.

In order to achieve competition goals and design objectives, 6 degrees of freedom is a mandatory requirement. Accordingly, ASUQTR's system engineers chose to have a balanced platform with an arrangement of 8 thrusters. This configuration facilitates stable movement control when coupled with a linear quadratic regulator (LQR) system. To improve reliability of control movements, a Dopler Velocity Log (DVL) sensor is used to monitor three position axes. This module was easily attached under the submarine in order because of the mechanical design choices. The choice of this specific sensor also simplified the system because only one module is necessary to produce data for 3 axes, where other design ideas required about 3 or 4 sensors for an equivalent reliability. The 3 remaining axes are monitored by a simple yet effective inertial measurement unit (IMU). The integration of this sensor with standard USB interface helped to achieve a desired level of reliability in very short times.

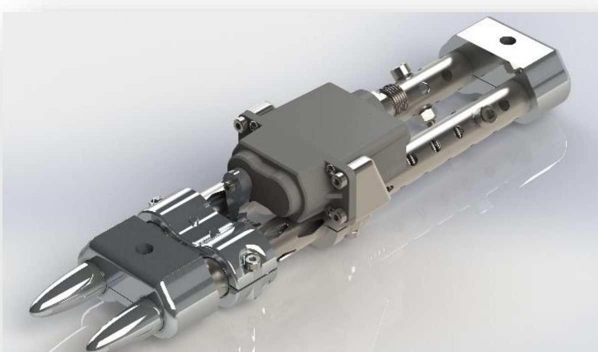
As a means of detecting pingers signals, ASUQTR's software team created a module from scratch including 1 FPGA and 5 hydrophones positioned in a square based pyramid. A real-time multilateration algorithm uses this arrangement to detect the position of pingers in 3 dimensions. The last module added on the AUV 2.0 platform is a Ping360 sonar from Blue Robotics to detect the location of challenges and obstacles such as walls. Like the other modules, the sonar is reliable (made by a third party), simple to integrate and fulfills the goal of diversity for the competition strategy since it is readily available for behavior development. The only exception (reasonably) to the competition strategy is the pinger detection module. Since the complexity inherent to the pinger challenge is hard to manage, this module does not have the reliability that a third party vendor could provide or the simplicity of integration of a USB sensor, but it certainly adds to the diversity aspect of the strategy, thereby allowing to attempt to score points for the pinger missions. ASUQTR could not afford a marketed solution for this problem, so its software team made use of a lot of creativity to implement a solution for 1/40 of the price of the marketed ones.

In order to have a reliable, simple and diverse system, ASUQTR's system engineers decided to add 4 embedded cameras, thus improving the capacity to accommodate changing needs of the AUV over the years. The first two cameras are positioned forward and used for object detection or feature recognition. Depending on the available resources during the runtime, it is possible to enable both of them for redundancy and distance estimation by means of stereoscopy. The third camera is placed downwards in order to perform feature recognition on the different paths used during the competition. The last one is installed on the gripper in order to have a maximum precision when grabbing an object. Once again, the implementation of those modules was possible because of the modular frame and the diversity-oriented design approach. This approach is also coherent with the competition strategy.

A custom gripper was designed and optimized to use the least material. A topology study has been run to find the optimal geometry. The gripper has been machined by ASUQTR's students from a block of aluminum, consequently it is reliable and simple to integrate and easy to validate. On the same topic, the support bracket of the camera pod located on the gripper assembly have gone through a finite element analysis (FEA) study to reduce its weight.



The torpedo launcher is a custom design that uses a servo motor as a trigger. The launcher also has two torpedoes centered under a camera to make it easier to aim. Both those characteristics make the launcher simple to integrate with the whole system and its function's reliability is improved by its chosen placement.

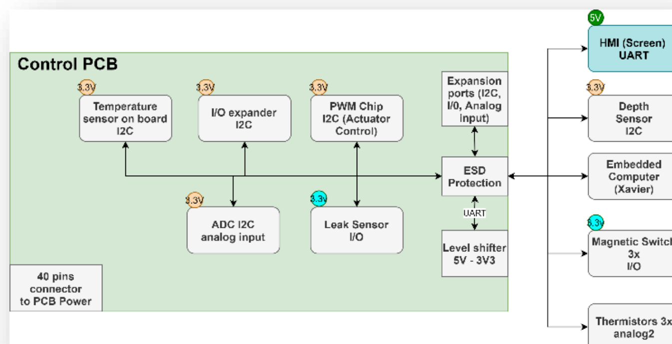


The dropper module is also homemade. A magnet located on the inside of the enclosure is pulling on another magnet on the outside of the enclosure until the inside magnet is rotated 180° apart by means of a servo. This action pushes out the magnet, assuring excellent repeatability. Thus, the design is yet again simple to integrate (only a PWM to command) and magnets are an easy to troubleshoot components.

Electrical

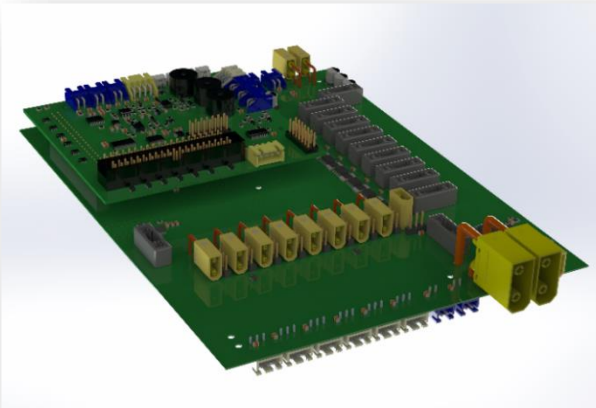
Requirements:

- Secure to manipulate for us and for the diver
- Easy to understand
- Easy fix if necessary
- Receive the voltage from the battery pack
- Generate every supply for each module of the AUV
- Power up 8 motors and drive them
- Communicate with the embedded computer
- Acquire, filter, and send hydrophone signal to the FPGA board



For AUV 2.0, the electrical team started from scratch for the design of the electronics. The idea was to have more flexibility with the choice of components and to ensure the reliability of these. The team also wanted something custom to fit all our needs, thus improving reliability by choosing components with durability in mind. With this idea, the architecture for electronic is divided into 4 boards: control, power, hydrophone and battery pod boards. This design decision improves modularity by separating functions, which in turns improves complexity management and reliability for upper layers of development.

The control board is used to control the actuators and sensors. It is also used to bridge the gap between the main computer and the sensors and actuators using I2C communication.



This board is a main component to achieve our diversity strategy because it allows the integration of additional sensors, if need be, which gives more potential solutions to solve problems for the software team.

The power board is used to distribute and convert electrical energy from the batteries to power all systems. It contains several voltage converters to supply the sensors with different voltage levels. Also, it allows to protect the systems from excessive current with the help of car fuses. An emergency switch is included to cut power to all the submarine's actuators. It is activated by a magnetic switch that can be pulled from outside the main tube. This PCB is the elementary component for reliability in the whole system. A lot of efforts went into its design to compensate for all kind of environmental events. This was done because to avoid undesired effects on energy supply of any component, independent of their specific voltage/current needs.

The hydrophone board is used to amplify, filter and route the signals from the hydrophones to the FPGA. This board is small and uses a lot of passive components because they are simple to implement the requirement and can achieve a greater level of reliability for a generic commercial solution of the same price.

The last board is the battery tubes (also called pods) board. In the electrical design, the batteries are separated in two different tubes to avoid the clutter of the main tube. Therefore, the battery tube board is printed twice and allows to give feedback of the batteries to the main computer via USB. The board contains an Arduino Nano because of the simple integration of this microcontroller for the software team. The board is also used to operate the magnetic droppers for the dropper mission.

All these electronic components have been designed to be compact so that all the electronics fit in the main tube and is manipulate. This way, the team will have the least amount of time possible related to the assembly of the electronics in competition and will be able to make changes quickly in case failure, thus improving the simplicity in the system.

Software

Requirements:

- Modularity in the subsystems
- Control node
- Vision node
- Hydrophone
- Behavior engine
- Easy to program behaviors
- Dashboard

Firstly, aims to achieve modularity for all subsystems. This facilitates teamwork and quick replacement of any module in case of failure or redesign. ROS (Robot Operating System) is used to implement each sub-module as separate nodes that receive and send messages to other components. Different layers of nodes were implemented to separate the hardware drivers from the logic, consequently improving simplicity and robustness. Each physical module has its own nodes to produce/consume data and execute error management specific to the module.

The control node passes data produced by sensors (DVL, IMU, target positions coming from behavior engine) through an algorithm to feed data to the consumers (the thrusters). This architecture design is very common, hence facilitating the management of its complexity for years to come and simplicity. This module uses an LQR algorithm because it is more straightforward to configure when the mechanical design changes compared to other control algorithms. This is mandatory to fit with our diversity competition strategy, because it implies to often add/modify hardware modules. Even if the config of the LQR is simpler to configure, it is nonetheless a complex controller to implement, because it takes into account the dynamics of the submarine, such as the placement of the engines and the forces that are exerted on the submarine (Coriolis force, water friction, gravity, etc.). With this controller, we can move in the 6 degrees of freedom simultaneously. Although the time to design is much longer, this solution was chosen because the ease of configuration saves a lot more time down the line. The simplicity of the configuration improves the reliability of the system in times of stress such as competition testing. With the cost function integrated in the controller, it is possible adjust the behavior of each motor (velocity or position). If the cost of a position is increased, the controller will try to hold that position more aggressively.

The vision node is essential for several challenges. This module acquires the various cameras with runtime configurable parameters which is very important to enable quick testing and simplicity of configuration. Due to the processing capacity, we cannot achieve object detection on more than one camera at the same time, on the other hand we can do feature recognition at the same time. The use of this module by the behavior engine is done by services in order to configure the objects to be detected, which camera to analyze and how. The feature recognition of this module is performed using OpenCV and allows the recognition of form (rectangle, square, circle) and color threshold. This module also embeds an object detection which uses a YoloV3 neural network with an accuracy of more than 90%, thus improving reliability. The last feature of this module is the use of stereoscopy to detect the distance of an object in short range (less than 2m).

The implementation of the calculations for pingner location is separate from the others because it is not performed on the main computer (Jetson Xavier), but rather on an FPGA. This eases the required computing on the main computer and allows to separate functions in a more reliable way. The calculation of the position of the pingers is carried out using a custom Goertzel's algorithm with a multilateration algorithm. This module is the only one programmed in VHDL in order to feed real-time data.

The behavior engine is the autonomous part of the software architecture. This module makes it possible to program actions and series of actions with infinite layers of encapsulation. However, for the sake of simplicity and maintainability, the maximum scope of encapsulation is a Robosub mission. This module is the manager of the main subsystems. Therefore, it takes decisions to use actuators depending on complexe sensor input (like the detection results of the vision node). This module is based on FlexBE behavior engine and allows easy graphical programming for everyone, even if they have no experience in code. This characteristic of the design decision is the the core of the simplicity competition strategy. Most of the software modules were made with the idea of cleanly and rapidly modifying autonomous behaviors without coding knowledge.

The AUSQTR dashboard module is an interactive front end designed to have real time feedback of what the software is doing when the AUV is executing missions. It also allows to directly interact with the submarine, like changing runtime configuration of the control system or current behavior.

III. EXPERIMENTAL RESULTS

So far, several tests and simulations have been done to verify the performance of the different systems of the submarine. The platform of the submarine has passed the basic mechanical tests and the watertightness tests to make sure there is no water leakage in the submarine. For the electronic part, we soldered and tested the control board and the power board. Both boards are working properly. We succeeded in integrating them into the submarine for the pool tests. The communication between the computer of the submarine and the electronic cards to control the motors works well and the emergency switch is operational. Afterwards, the integration of the software went well with the electronic part. Thanks to the communication between the two teams, we were able to make the submarine work in the pool on the first try this year despite the new electronic parts. For the control tests, the LQR control algorithm was simulated using Matlab-Simulink. We needed to simulate the algorithm to verify the feasibility, accuracy and robustness of this type of controller. We modeled the submarine in equations to see how the controller reacts on the 6 DOFs of the submarine in Simulink and to see what we should expect during our physical tests.

In simulation, the submarine managed to reach the objectives set in the simulation with sea currents of 1m/s. For the pool part, we tested the control of the angles, and the submarine is able to maintain the angles very well. The only part that remains to be tested is the control of the movements of the submarine which will be tested once the DVL is installed on the submarine, but so far, the model agrees with the simulations. Finally, the object detection system has been tested in the pool and everything works as expected. Now that we have a submarine with the functional base, we continue our tests to implement more complex functions for the competition. Many hours of testing are still needed to pass most of the challenges, but with our strong team we will succeed.

IV. ACKNOWLEDGEMENTS

A huge thank you to all our sponsors without whom, the project would be impossible to realize and to all our members who put a lot of time into the project. In this way we can all grow as future engineers.

Thanks to Trois-Rivières engineering, Atlassian, Industrial Electronics Research Group, IEEE, LSSI, Solidworks, NVIDIA, Kongsberg Automotive, VectorNav, Martin Perreault, Matlab, LabelBox, GEGI, WaterLinked, Novo, Altium Designer, GEGI student branch, Trois-Rivières engineering school, BlueRobotics, Aquarian and all the professors who support us.

References

https://opencv.org/. (s.d.).
https://www.ros.org/. (s.d.).

Appendix A: Component Specifications

Component	Vendor	Model/Type	Specs	Cost
Buoyancy Control	• Blue Robotics	• Stainless steel ballast x 8 • Subsea Buoyancy Foam R-3312	• 7 oz • 16"x 8"x 1"	• 72 $ • 41 $
Frame	• Custom	• SLS 3D printing • Acrylic sheets	• NYLON 12 • 1/2"	• Donated • Donated
Waterproof Housing	• Blue Robotics	• 8" enclosure • 4" enclosure x 2	• 12" • 12"	• 432 $
Waterproof Connectors	• MacArtney • Blue Robotics • Blue Robotics	• DBH8F/DIL8M • Cable Penetrator x 25 • Cable Penetrator x 8	• 8 pins • 6 mm • 8 mm	• 500 $ • 125 $ • 40 $
Thrusters	• Blue Robotics	• T200 x 8	• Max thrust: 11.2 lbf	
Motor Control	• Blue Robotics	• Basic ESC x 8	• 30A brushless ESC	
High Level Control	• Custom control PCB	• PCA9685 IC • JLCPCB	• 16 channels PWM output	• 200 $
Actuators	• Blue Trail	• Servo SER-120X	• 34.0 kg-cm	• 480 $
Propellers	• Blue Robotics	• T200 x 8	• 3" diameter	
Battery	• Venom Power	• SKU 35005	• 13AH 4S 14.8V 15C	
Converter	• Walfront	• Wal front4bnaie7oy3 2x	• 10A Max • 150kHz	• 18 $
Regulator	• Custom PCB	• LM3150 IC • JLCPCB	• 10A Max	• 500$

Component	Vendor	Model/Type	Specs	Cost
CPU	• NVIDIA	• Jetson	256-Core NVIDIA Pascal GPU, Dual-Core NVIDIA Denver 64-Bit CPU	
FPGA	• Digilent	• ARTY Z7-20 ZYNQ 7020		• 278 $
Internal Comm Network and External Comm Interface	• Black Box	• LBS005	10/100-Mbps Copper RJ45, USB Powered	• 60 $
Inertial Measurement Unit (IMU) / Compass	• Vectornav	• VN-100	3-axis accelerometers, gyros, and magnetometers, a barometric pressure sensor and a 32-bit processor	• 800 $
Doppler Velocity Log (DVL)	• WatrerLinked	• A50	• Ethernet Communication • Velocity resolution 0,1mm/s	• 6000 $
Vision	• Teledyne Flir	• FFY-U3-16S2C-C-DL x 4	1.6 MP Color Firefly S USB3 Deep Learning Camera, Sony IMX296 1/3" CMOS, C-Mount	• 1600 $
Acoustics	• Blue Robotic • Aquarian Audio	• Ping 360 • Hydrophone	• USB communication • H1A	• 1975 $ • 800 $
Manipulator	• Blue Robotics	• Newton Subsea Gripper • Custom aluminum claws	• 22lbf grip force	• 439 $

Component	Vendor	Model/Type	Specs	Cost
Algorithms: vision	• YoloV3 • Detecnet			
Algorithms: acoustics	• Goertzel (Custom)			
Algorithms: localization and mapping	• Image recognition • Sonar detection			
Algorithms: autonomy	• FlexBee			
Open source software	• OpenCV • ROS			
Team Size (number of people)	• 10			
Expertise ratio (hardware vs. software)	• 6/4			
Testing time: Simulation	• 80 hours			
Testing time: in-water	• 100 hours			
Inter-Vehicule communication	• N/A			
Programming Language(s)	• Python			

APPENDIX B: SOFTWARE DESIGN NETWORK