FEFU & IMTP AUV Team RoboSub 2021 Autonomous Underwater Vehicle

Vadim Saburov, Artem Lisovitckii, Konstantin Shilin,
Alexander Torzhkov, Stepan Egunov, Dmitriy Stabrovskiy
Far Eastern Federal University 8, Sukhanova str., 690950, Vladivostok, Russia
Email: fefu.impt@gmail.com
Web-site: www.fefu-imtp.info

***Abstract*: The following paper describes the improvement of the AUV which our team has prepared for RoboSub 2021. The main goals of our work are participation in the competition, research, and development of technology in grabbing objects and improving our vised-based navigation software. The paper describes and justifies our solutions in developing a new vehicle, including its shape and materials, software solutions, and a strategy named "More mechanics, more reliability."**

I. COMPETITION STRATEGY

Our team has been participating in RoboSub since 2012, but in 2020 we couldn't take part in the competition to our great regret. Now with lessons from the past, we started to consider new ways in mechanics and software. Since we have a limited number of team members, we still prefer the evolutionary path of development of the vehicle, wherein only one part of the AUV is significantly changing at a time. Our main achievement of this year is radically modified mechanics. In fact, we don't have enough time, and our strategy in these circumstances was a few changes and reconfiguring the vehicle under the new rules because Russians love to say, "Don't reinvent the wheel." This means that we are focusing on the main problems that we had in past competitions.

The problem was with grabbing items and setting up the software. Our vehicle is capable of grabbing items using a "slap" bracelet. That genius idea came to the mind of our teammate because he saw how a child brought this bracelet and played with him. The main idea of this mechanism is that when it touches the bottle, the center of the bracelet is under pressure and goes from open to the closed state while grabbing the bottle. Why do we choose such a simple mechanical system? Because from our experience we have a result that the electronic system is too unpredictable and our observation in the torpedo system gives a good result.

We have several approaches with a grabber to solve the problem of releasing objects. The first approach is to throw the entire grabber mechanism entirely. And we hope that the penalty for this action is less than the number of points that we can obtain for surfacing up at the octagon with an object. The second approach is not to drop off the object and carry it with AUV throughout the whole final performance. We will hope that the octagon will be the last task in a random pinger case in the third case. This option is the most convenient since the object does not need to be dropped, and the kind diver may do it for us.

We can finally tell the secret of our software, which has been stored for 5 years. We practically do not program the robot but simply list tasks for it to perform. For example, in our “mission tasks,” we set the object’s position and tell the vehicle to go to this point with specific parameters.

Traditionally we have several plans. In the worst situation, we use our well-known strategy, “Vorota -> Pinger” (Vorota means Gates in russian). This strategy is based on the unusual hull design that allowed us to get through the gate without any machine vision. “Pinger” from “Vorota -> Pinger” means that we rely on our acoustic to hear pingers and go to them, all fast. However, we are optimistic people, so our team has a plan “maximum,” which means successfully completing all tasks with a true random-pinger. And our final plan is to “determine” a random pinger that is like a “maximum” plan, but in this situation, ms. Fortune is not on our side, and we refused random scores and asked for a certain sequence of actions, including starting the mission.

II. DESIGN CREATIVITY

A. General construction

As said earlier, we didn’t change the vehicle shape, and so the design advantages from Robosub 2019 were saved. Our main idea in the vehicle design was to “make it as light and small as possible, so that one person could carry it, and so that it could fit into one suitcase”. As a result, we created vehicles (figure 1) with huge volumes for a large number of solutions. These advantages are:

1. One of the lightest and smallest vehicles.
2. Has 2 stereo cameras, shooters, acoustics, and new grabbers
3. Has a narrow shape with small frontal resistance that makes it easier to move in a straight line
4. Although it has only four thrusters, it can move laterally, as well as longitudinally, vertically and by yaw.

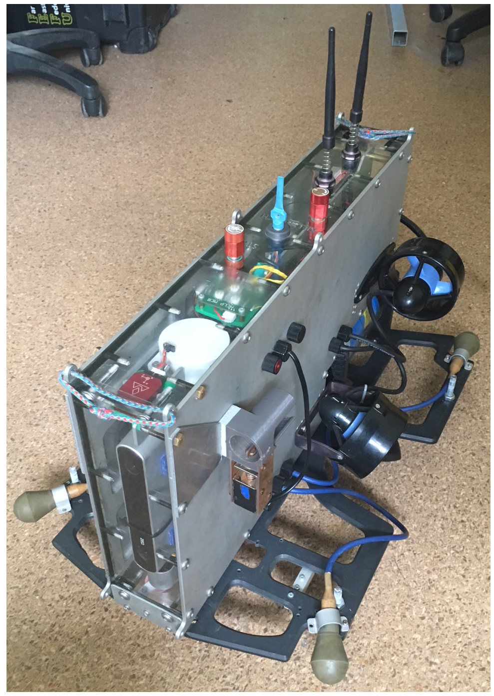

Figure 1: The Pandora vehicle.

However, the manufacturability of assembly is not as good as wanted for several reasons: the difficulty of construction, specific forms that give complexity in assembly, and a caring attitude to the vehicle.

B.Mechanical

In Robosub 2019, we solved a problem with shooting. However we still had a problem with grabbing. This year we focused on this problem, but the solution to the problem wasn't found until one of the days our member saw a child who played with a “slap” bracelet (figure 2). Now we have another one in our vehicle because we've known from experience that the best ideas are the simplest things. This bracelet is a spring that has two stable states. In its long, straight shape, it is stable but with a strain on it. It makes it want to bend into a circle. Deforming it a little bit by applying pressure to one part causes that part to roll up and reduces the strength to maintain its shape, causing the rest to roll up in a chain reaction. Pandora still equipped a torpedo that we created in 2019 because the main advantages that these mechanical devices and effectiveness underwater still saved.

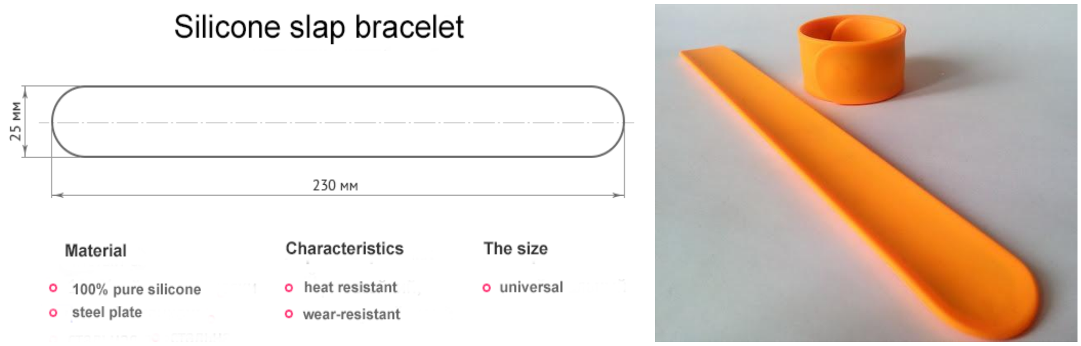

Figure 2: Silicone slap bracelet for grab mechanism.

The hardware of our video system consists of two Stereolabs Zed cameras. We deliberately used the large cameras, because the box-shaped Pandora allows us to place them all. In turn, the increased stereo base of the camera (for example, in comparison with the Zed-Mini camera), in theory, makes it possible to determine the distances to objects more accurately and reconstruct a 3D point cloud confidently.

C. Electrical

One of the problems of any AUV is the need to indicate its state and problems during operation. Since we have a “Pandora's box” - it entails a source of significant and unexpected troubles. Therefore, we decided to "hire" a security officer, and for this, we need to equip a problem display room in our vehicle.

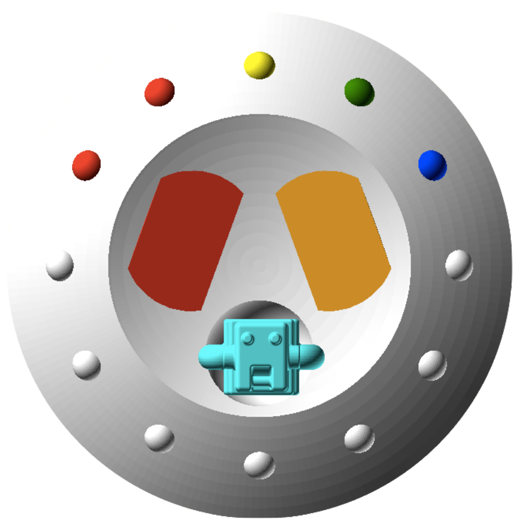

Figure 3: New Pandora’s display panel 3D model (top view).

Our display panel looks like a wall clock (figure 3), allowing the officer to communicate problems quickly and efficiently. We have 2 red indicators that give information about pressure in the vehicle and leakage, one yellow about power, one green that gives information of mission start, one blue led indicator about details of the mission, and 7 white LEDs that provide additional information (for example 2 for shooting and so on). For this task, we decided to create a PCB with transistors, which switches these LEDs. Initially, transistors were to be controlled by I2C->GPIO custom converter, but we came across a problem with unstable work of the I2C interface on the Auvidea J120 carrier board. To solve this problem we decided to change our I2C->GPIO module to manufactured USB->GPIO. In addition, on the proposed version, you can use half of the pins as the inputs, even analog.

D. Software

A distinctive feature of our software is the complete absence of the client part. No pre-installed software is required on the operator's computer, because all programs are located on the vehicle. Such an approach allows you to use any laptop, tablet or smartphone to launch a mission and control Pandora. The complete description of our software is much more than this report. Let's highlight several main areas:

1) **Software platform**: We use a software platform that is not standard for Robosub teams. After a year of experience with ROS, we chose the "russian alternative" of this system[https://www.researchgate.net/publication/312288175_Reconfigurable_distributed_software_platform_for_a_group_of_UUVs_yet_another_robot_platform], developed at the Institute of Marine Technology Problems. This system is a real-time reconfigurable software platform designed for data exchange between software modules of unmanned underwater and surface vehicles. It is realized as a light-weight OS-independent library that allows integrating AUVs, ROVs and surface vehicles to a single computing cluster. The advantage of this system is the possibility of decentralized information interaction through the low-speed and unstable channels. This platform does not require any additional tools or software modules installed to the robot control or operation systems. All that is required is a single library for interprocess communication, data logging, and a web-based graphical user interface (GUI). The system used is very flexible. For example, it allowed us to deploy on a remote computer our stripped-down control system for several simultaneously online-controlled virtual AUVs [http://viewer.fefu-imtp.info/].

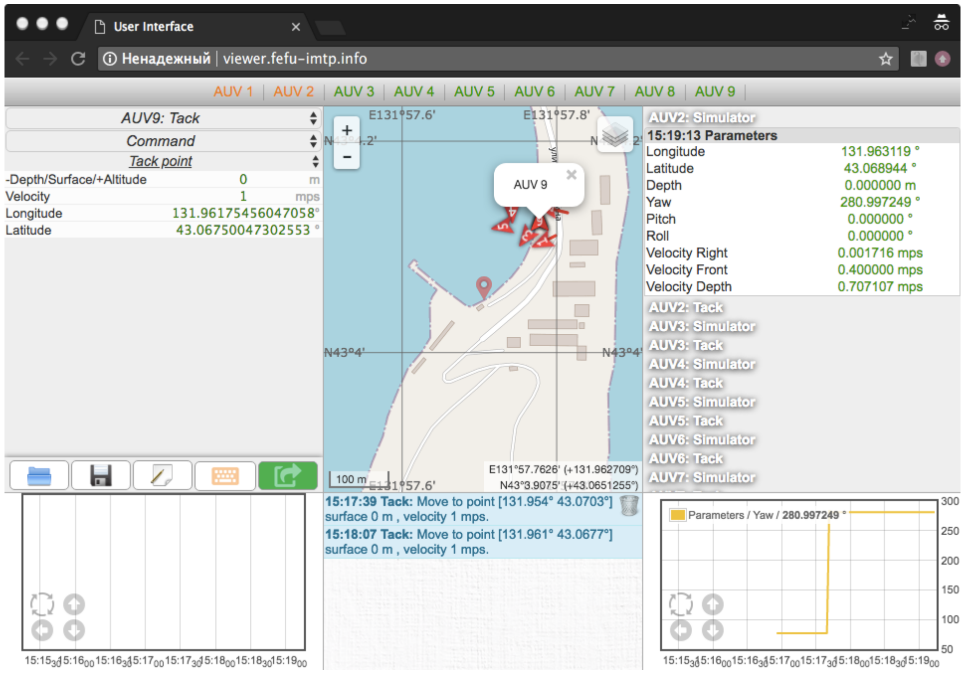

Figure 4: Web-based GUI of used software platform.

The GUI of this platform (figure 4) is based on the mechanism of web sockets and uses JSON components for data verification. Thanks to it, we can monitor the status of all the modules of our vehicle and calibrate them. In addition, with its help, we can easily and safely create missions. Safety, in this case, lies in the fact that we do not need to make changes to the code, and the started mission is guaranteed to be syntactically correct.

2) **Mission tasks**: The robot's mission looks like a list of tasks. Each task is an action (like “drop the ball”, “fire the torpedo” etc.) or AUV movement. All tasks are visible in our WEB-based interface (figure 4) and so “in the head” of our vehicle (figure 5 - only one of the tasks).

```
{
    "id: "Stabilization",                  // Command ID
    "value[Reference]": "Transponder",     // Reference to coordinate system
    "value[Right]": 0,                     // Right offset form Reference, m
    "value[Forward]": 0,                   // Forward offset form Reference, m
    "value[Depth]": 3,                     // Depth of stabilization, m
    "value[Time]": 0.1                     // Time of stabilization, minutes
}
```

Figure 5: An example of the mission task representation in Pandor's mind.

Pay attention to the "world of our vehicle" is absolutely deterministic and cannot have syntax errors due to JSON-schema used for mission description [https://json-schema.org/]. Even our "random pinger" is an unavoidable consequence of the determinism of the model we are using. The whole point of our mission is that we have many systems of coordination and jumping between them. We have a global coordinate system, coordinate system of mission (at the start it lies in current AUV global coordinates and turned in the direction of AUV), and coordinate systems of all of the objects. For example suppose we need AUV to approach the hydroacoustic beacon. Task code will look at the JSON language as on figure 5 (text after "//" is a comment not supported by JSON-standard and used for explanation only).

3) **Video system**: In our software we use quantized SSD Mobilenet with Tensorflow Lite for object recognition. Our team decided to go from TensorFlow to TensorFlow Lite to minimize resource usage, size of memory, get more FPS, and get more stable code. Also we set a goal to turn on two stereo cameras on one computer for the purpose of decreasing equipment. Our first solution with the J120 board gives a negative result. And the problem lies not in the processing algorithms, but in the lack of power. So we use a USB3.0 hub with an external power supply. Now we can get images from two cameras using only one computer.

III. EXPERIMENTAL RESULTS

We have carried out a number of field tests with AUV, mainly for testing new systems. Experimental results and experience of exploitation are as follows:

Figure 6: Experiments with torpedos (left image) and grabbers (right image).

1. The main problem of the developed grabber (figure 6 - right image) was in the twisting of the mechanism itself because it is unbalanced and twists into a cylinder. We tried many variants of the bracelet and, in the end, fixed two magnets at the ends of the bracelet that gives a less radius of twisting and more

reliability in the lifting load. In addition, we found some problems with the grabbing of a real bottle (not a PVC construction) from the seabed due to the grabber claws resting on the bottom. But lifting a real bottle (not made of PVC), as we understand it, is not included in the competition plans.

2. As mentioned before, we tried to create a power switch using I2C but met with freezing the channel at the worst possible moment (Final of the SAUVC 2019). Therefore, we switched to the GPIO of the J120 expansion board, but ran into problems when initializing the OS (switches its GPIO pins randomly and shoots all our torpedoes and drops balls). So we made an appropriate conclusion and went to the USB->GPIO adapter.
3. Magnetic compass. We tried to use different magnetic compasses (Xsense, VectorNav) in our vehicle but had problems with AUV direction at the most inopportune moment. Because the pool has a lot of metal, the magnetic compass became unreliable. Today we use the Fizoptika-VG1003 fiber-optic gyroscope from the Russian company Fizoptika [https://fizoptika.com], which allowed us to solve the problems with the yaw of AUV for quite little money.
4. Experience and troubles with a fragile acrylic frame. When screwing bolts into acrylic frame you can create too much tension, and this gives a crack. We decided to replace the bolt with a threaded rod and put it to the frame with glue. No tension is applied to acrylic now. If you want to use something similar to Pandora's frame technique, just use a different mounting method then 50 bolts.

ACKNOWLEDGMENT

The development of our team is supported by Far Eastern Federal University [https://www.dvfu.ru/], Institute of Marine Technology Problems of Far Eastern Branch of Russian Academy of Sciences [http://www.imtp.febras.ru/], MARINET [https://marinet.org], NTI [https://nti.one/]. We would like to thank our mentors and advisors: Alexander Pavin, Grigorii Eliseenko, Vitaliy Storozhenko, Andrey Makhliarchuk, Pavel Efimov, Anton Tolstonogov and those from FEFU and IMTP who have supported us in this work.

Appendix A Component Specifications

Component	Vendor	Model/Type	Specs	Cost (if new)	Status
Frame	Developed by team	Plastic			Installed
Waterproof Housing	Developed by team	Acrylic glass and aluminum			Installed
Thrusters	Blue Robotics	T200	https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/		Installed
Motor Control	Zubax Robotics	Orel 20	https://files.zubax.com/products/io.px4.sapog/Zubax_Orel_20_Datasheet.pdf		Installed
Actuators	Developed by team				Installed
Battery	Developed by team	Li-Ion 18650			Installed
Regulator	Texas Instruments	TPS5430	https://www.ti.com/lit/ds/symlink/tps5430.pdf		Installed
CPU	Nvidia	Jetson TX2	https://developer.nvidia.com/embedded/jetson-tx2		Installed
Internal Comm Network	Ethernet		1000 Mbps		Installed
External Comm Interface	Wi-Fi		100 Mbps tether or		Installed

			Wi-Fi		
Angular Velocity Sensor	Fizoptika	VG1003	https://www.fizoptika.ru/download/19d34f47b701288c602774db75628469.pdf		Installed
Inertial Measurement Unit (IMU)	VectorNav	VN-300	https://msmedia.surf/products/detail/vn-300?__cpo=aHR0cHM6Ly93d3cudmVjdG9ybmF2LmNvbQ		Installed
Vision	Stereolabs	Zed	https://www.stereolabs.com/zed/		Installed
Acoustics	Aquatelecom	AT-2045	https://aquatele.com/products.html		Installed
Algorithms: vision	Tensorflow	Lite			Installed
Algorithms: acoustics	Developed by team		Signal delay, FFT, Hilbert transform		Installed
Algorithms: localization and mapping	Developed by team		Kalman filter, Hough Transform		Installed
Algorithms: autonomy	Developed by team		Multiprocessing		Installed
Open source software	Leaflet, Echarts, Flot Charts JSON Editor		www.leafletjs.com/ www.echarts.apache.org/ www.flotcharts.org/ www.github.com/jdorn/json-editor		Installed

Team Size (number of people)	6 team members and 3 advisors				
Expertise ratio (hardware vs. software)	50/50				
Testing time: simulation	100				
Testing time: in-water	100				
Programming Language(s)	C++, JavaScript				