

Implementation of a low-cost autonomous underwater vehicle using open source ROS components with consumer class sonar technologies

T. Aaltonen*, M. Saarivirta**, T. Kerminen* and J. Grönman**

* Satakunta University of Applied Sciences, Pori, Finland

** Tampere University/Faculty of Information Technology and Communication Sciences, Pori, Finland
toni.aaltonen@samk.fi

Abstract - The increase in the availability of low-cost components has made it possible to design and implement new and innovative devices for the underwater robotics. Today, cost-effective and open solutions can be created to replace previous expensive and proprietary robot systems. This study presents the prototype of an autonomous underwater vehicle. The system was built using commonly available sensor components in combination with an open source code and in-house designed parts. The configuration of the autonomous underwater vehicle is introduced in this study. The results obtained from the component tests performed in the actual operating environment are presented. Based on these tests, the subsystems were found to be well-suited for underwater environment.

Keywords - AUV; low-cost; ROS; sonar; open-source; underwater robotics

I. INTRODUCTION

Autonomous Underwater Vehicles (AUV) have been subject to scientific interest for some time, and a continuum of research projects have been developed around the theme, such as the European ARCHEOSUB project. [1] Interest in AUVs has also awakened in the business world; robots are now increasingly used in commercial sites, such as oil and gas industry, as well as fish farming. [2] The increased use of AUV equipment is supported by their low operating costs and the reduction of personnel risks in challenging conditions.

Satakunta region is a highly industrialized province which Rauma-Repola Oceanics established already in the 1980s by delivering a technically advanced Mir submarine to the Soviet Academy of Sciences to study the secrets of the deep sea. Ever since, there has been developed a cluster of companies focusing on automation and robotics in the Satakunta region. The starting point for this research was to combine those two high tech engineering sectors, maritime and robotics.

The aim of the study was to create a low cost AUV based on commercial off-the-shelf and open source components. The design of the prototype largely takes advantage of the solutions created by the open source community and the literature in the field. The AUV was

built in RoboAI Research and Development Centre founded by Satakunta University of Applied Sciences (SAMK) and Tampere University. This RoboAI study was a joint project for SAMK's Automation research team and Tampere University's Data Analytics and Optimization research group.

II. BACKGROUND

The first SPURV (Self-Propelled Underwater Research Vehicle) was built in 1957 at the University of Washington, USA. It was made of aluminum and resembled a tornado. It could dive up to 3600 m and was able to travel at a speed of 4 to 5 knots. Its primary power source was the propulsion battery with two sets of 16 silver-zinc cells connected in parallel through diodes. The battery was 24 volts and 200 Ah. Secondary power supply was provided by four solid-state converters. [3] SPURV II was developed in 1973 with a greater range, speed and operating time. The SPURV II could be operated by a computer stationed on a marine vessel. Both were still operated until the 80's. [4] In 1983, ISE (International Submarine Engineering) Ltd. developed in co-operation with the Canadian Hydrographic Service (CHS) and the Department of Defense the ARCS (Autonomous and Remote-Controlled Submarine) as a test platform for autonomous submarines. It has been used since 1987 for development and demonstration of autonomous underwater technologies. Today, ARCS is used to develop new AUV technologies e.g. navigation system, algorithms and battery types. [5] In 2001, Woods Hole Oceanographic Institute Laboratory developed REMUS (Remote Environmental Monitoring Units). It was intended for marine scientific community for precision tasks in coastal areas. Its small size made it relatively unique as it was only 19 cm in diameter and weighed only 37 kg. Having four lithium-ion batteries it was capable of a speed of three knots with a range of 80 km. It was designed to be operated with a laptop. [6]

The Robot Operating System (ROS) was developed by the Stanford Artificial Intelligence Laboratory (SAIL) in 2007. ROS is becoming a very popular open-source middleware system (i. e. collection of software frameworks) and it is widely accepted by the research and industrial communities. Its goal is to make robotics system

easier to develop and share the work on other robots with minimal changes. Meaning that the code can be reused and improved by many different users and platforms. [7] ROS provides a set of tools and libraries. It supports message-passing between different processes that run on one or several computers. Processes are called “publishing”, “subscribing” and “messages”. Both publishing and subscribing to a message are called “nodes”. Data logging is also possible and a UUV (Unmanned Underwater Vehicle) Simulator package is available on GitHub containing Gazebo plugins. [8] Among others, ROS can be found in drones, service robots and AUVs. The future potential of ROS lies in the expanding applications of robotics and the developing of new, smarter applications in robotics.

RoboNation’s international competition, Robosub gathers student teams from both universities and high schools around the world to challenge each other in the making and developing of AUVs.

In the USA, Riverside’s student organization, RoboSub UCR from the University of California, has built an AUV called Seadragon. It is capable of completing several aquatic tasks on its own. They are using ROS, Python, C/C++ for their controls. [9] The University of Florida (UF) presents its eighth generation SubjuGator that is developed by undergraduate students in UF’s Machine Intelligence Laboratory (MIL). The university has a 22-year history of autonomous vehicles. The newest SubjuGator concentrates on robust control, hardware improvements and software innovations. [10] At Michigan Technological University the Nonlinear and Autonomous Systems Lab (NASLab) has developed and combined a backseat controller for the General Dynamics Bluefin SandShark AUV. The integration of ROS is important to promote transference of code and skillset across domains. They have installed ROS on the Raspberry Pi with written drivers to initialize communication. [11]

In Spain, Underwater Robotics Research Centre (CIRS) of the University of Girona developed and tested the SPARUS II AUV. It is a lightweight classic torpedo-shaped vehicle with a hovering capability and a mission-specific payload. The software is based on ROS for multipurpose applications. [12]

Other parties have also taken interest in developing AUVs, such is a national project of Japan named “Next Generation Technology for Ocean Resources Exploration”, its nickname is “Zipangu-in-the-Ocean”. HOBALIN, the hovering AUV was originally developed in 2015. However, because it was based on an old design, they decided to replace the software with a new design based on ROS. HOBALIN completed its survey missions on February 2018. The authors continue improving the AUV. [13] Hydroid (USA), owned by Kongsberg Maritime (Norway) has announced that the “Next Generation” of the REMUS 100 will be using ROS interface. Currently no open source is available. [14]

The University of Tampere has also given its own contribution to the AUV development. In 2016 the UNEXMIN project (funded by European Union’s Horizon 2020) produced a high performance AUV, named UX-1 for the inspection of abandoned flooded mines. The UX-1 can

dive up to 500 meters. Its spherical form with a 30 cm radius and 8 thrusters is well suited for the inspection of narrow mine channels. [15] Later on, in 2019 they have improved the propulsion system of the UX-1 by developing creative subsystems to improve the mobility of the robot. [16]

III. IMPLEMENTATION OF AUV

The SubjuGator project from the University of Florida was used as an example for the project. The University has more than 20 years of experience in AUVs and it has won the Robosub competition 3 times and been in the top 3 for 7 times. The vehicle is based on ROS. This in all was a good starting point. Although there came to be some major differences. Mainly because of the differences in the budget. The SubjuGator’s sonar alone cost ca 60 000 € and the software package made 7 000 €. In addition, the Doppler Velocity Log (DVL) used to measure the vehicles speed was around 20 000€.

The second major difference is that the Robosub competition is organized in a test pool. In clear water, camera-based machine vision systems are a cost-effective and easy way to identify objects. The vehicle of this project is intended to operate in the Finnish murky waters where the possibilities of using cameras are very limited. Visibility in coastal waters are often reduced to a few dozens of centimeters and may be limited to as much as twenty centimeters. The thruster system of the project also became quite different.

A. Hull structure

The frame of the vehicle was made from aluminum profile. Two pressurized hulls were placed for the electronics. The sonar unit and the main computer were placed in the larger upper hull whereas power supplies, batteries and motor-drives were placed in the lower hull. The device was designed to be slightly buoyant so that it automatically returns to the surface in case of a failure. The choice between remote access and battery operation is made by changing the flange of the lower pressure housing. One of them is a solid flange and the other one has electrical and optical cable where the optical cable acts as a telecommunication link, and due to its tensile strength, as a traction protection for the electric cable. In addition, a pressurized air hose runs in the same bundle, the purpose of which is to make the cable slightly buoyant so that it does not stick to the shapes of the base. At the same time, buoyancy also reduces the resistance to the device.

B. Computer

The main computer was an AMD-based x86 computer. The processor was the Ryzen 2700 with 8 cores and 16 threads. The motherboard was Gigabyte B450 mATX with 2 memory slots. The main memory was 32 GB of DDR4. The graphics card was Asus Dual GTX 1070 with 8 GB of memory. The hard drive was a Samsung EVO 970 512GB NVMe SSD. The computer was powered by HDPLEX

400W DC-DC. This can be used both with current from the batteries and with the 24v DC voltage of the main power supply. The main power source which converts 240 Vac ground power to 24 Vdc was the Meanwell DPU-3200. Its efficiency was promised to be over 94% even at full 3192W power.

C. Sonars

Robotic sonar units will cost orders of magnitude more than could be procured with the budget for this project. As a result, less conventional alternatives had to be sought. The sonar used in yachting is quite inexpensive, but still quite good in image quality. In addition, they are easy to use. The problem with these is that they are not intended for such use and as a result, imaging on a computer is challenging.

This project went through various manufacturers such as Hummingbird, Lowrance, Raymarine and Garmin, looking for alternatives. It was then found that whereas the older serial fieldbuses for yachting were relatively open and used by all, such as NMEA0183 and newer NMEA2000, Ethernet-based busses are the manufacturer's own closed ones and only discuss between the manufacturer's own devices. It was not even known whether the traffic was encrypted in any way or not. The data produced by modern sonar is so accurate and bandwidth consuming that all manufacturer's sonar data travels between devices on Ethernet-based buses. There are several ways to get data from a device. The Ethernet protocol used can be determined by listening to traffic and writing a program that uses this, which appears as a manufacturer's device on the network, so that the sonar can send its data to it. The problem here is that at least two sonar units are needed to talk to each other over the Ethernet-based bus in order to listen to Ethernet traffic.

Another option is one sonar unit and some other device that uses, for example, an Ethernet-based sonar sensor on the bus. Another problem is that there is no certainty whether the traffic is encrypted or not. With all these things in mind, an alternative way of getting data on a computer was considered. Some of the better model series from Simrad and Garmin have an HDMI output. HDMI-USB3 capture cards for video processing, recording and retransmission are available in the price range of less than 100 €. For this reason, it could be assumed that the delay would be reasonable. The degree of difficulty in the implementation of this solution is considerably lower than in the determination of the bus protocol. Even if the accuracy is a bit off, it is worth the saved efforts as well as the reduced risk. For these reasons, this solution was reached. At the time of the decision there was no certainty about the functionality as there was found no prior information of such ever been done. The sonar was selected as Garmin's 8400 Xsv and the sensors were Lvs12 real-time sensors and GT54UHD-TM side-scan sonar. The solution has worked as expected, and there has been no visible delay between the image of the sonar unit and the image transferred to the computer. The effect on image quality has also proved to be insignificant.

D. Location and velocity information

The underwater depth information can be divided into two parts, the distance from the bottom and the distance from the surface. The distance to the surface is simply obtained by pressure measurement. MS580314BA01-00 was selected as the sensor. According to the sensor manufacturer's data sheet, the accuracy should be maintained within 30 cm even for longer periods of time.

The distance to the bottom is obtained by the sonar unit via the NMEA2000 bus. There has not been found any precise information of the accuracy of this, but when tested a few times with a measuring stick in the data gathering phase, the accuracy appears to be of dozens of centimetres. Of course, different bottom shapes can affect this.

The IMU, or inertia measurement unit is a commonly used sensor. The prices of these start from a couple of euros and the most expensive ones are tens of thousands of euros. In these, short-term information is usually useful, but sensor errors accumulate rapidly so that they cannot be relied upon alone, especially when calculating velocity and position from acceleration.

The weakness in calculating velocity from the sonar image is its dependence on the type of the bottom. There is no constant shape at the bottom from which the velocity could be calculated. On the other hand, there are some bottom types that the speed is available almost all the time.

At its best, the impeller only measures the flow of the water and only the velocity can be calculated in relation to the water. This can correlate well with the ground speed in certain situations, for example in small lakes, but in places where the water flows, the speed obtained in this manner is not reliable. The sensor is also easily broken down. However, this feature can be improved by installing more sensors. The impeller sensor is used to detect transverse motion. GPS location information has no cumulative error problems but does not work under water. GPS can be used as an occasional ground truth, by visiting the surface.

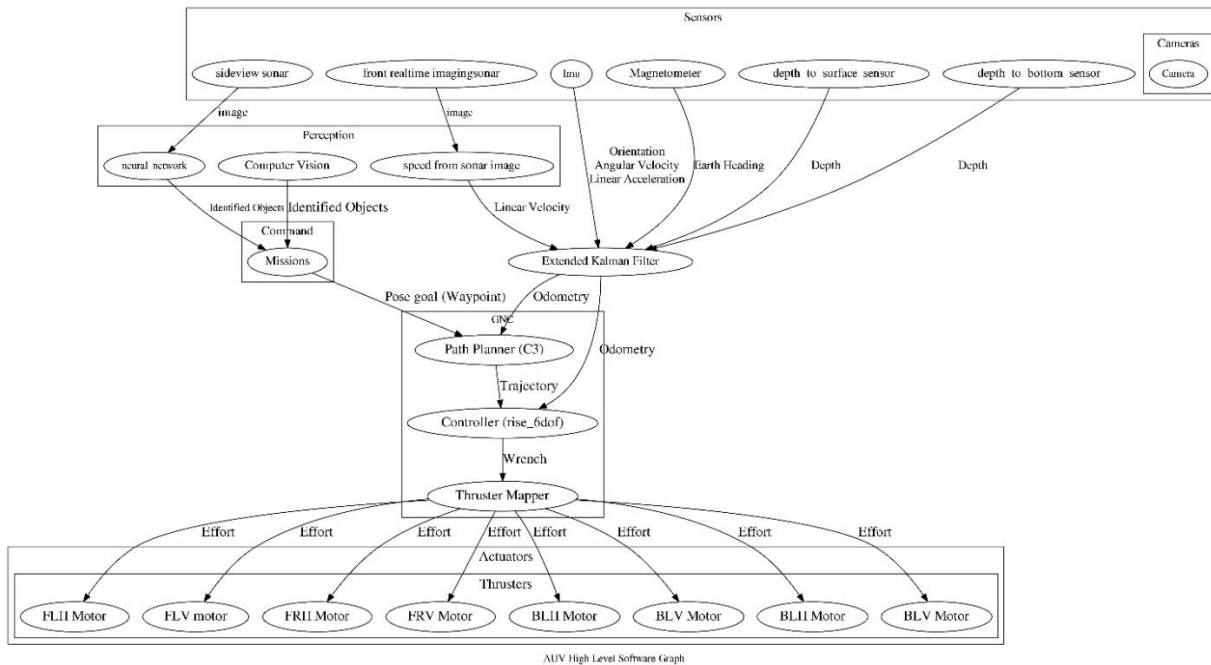


Figure 1. High level software architecture

IV. TEST PROCESS

E. Thrusters

There are eight thrusters and they are positioned as shown in Fig. 3. The structure decision was made because even if any of the thrusters were damaged, the task could still be completed, and the movement would still be successful in any direction. In some cases, the failure of either thrusters does not yet restrict the mobility of the device. Excluding the motors, the thruster assembly was 3D printed with PLA (polylactic acid) filament. Motors in thrusters are brushless DC motors normally used in RC helicopters. Each motor can be operated around 840W continuous power. Small zinc plates are attached to motors, for enhancing corrosion resistance. The thruster motor controllers are slightly modified VESC - open source ESC-controllers that have an RS485 bus to connect them to the computer.

F. Software architecture

In Fig. 1., there is a high-level architectural image of how information moves in a ROS system. The architecture is largely borrowed from the SubjuGator project that has been modified to fit the project. The mission system, path planner, controller, and thruster mapper are nearly identical to those of the SubjuGator. The Kalman filter is a robot localization package. Sonars generate data of detected objects and possible obstacles. The mission planner tells you the desired location. The sensors are collected on an extended Kalman filter, which provides the information and the desired location to create the desired route. It is then calculated how much power is desired for each motor so that the desired motion occurs. The Rise Controller monitors that the desired movement occurs.

Data gathering was carried out by attaching the sensors to various boats and the sensor data was stored on a laptop computer. The data gathering operation had several different objectives. There was the desire to learn how to calibrate and use sonars and to test sensors and bus software. The test uses ROS nodes and gathered data, as well as gathering data for teaching neural networks and calibrating algorithms. The most significant consideration for neural network-based velocity calculation is that GPS speeds which are not available below the surface of the water can be recorded while driving on the surface. In the data acquisition hardware, the ROS software ran on a laptop to which the devices were connected via USB buses. The devices are installed in the housing of the Fidobox. The sensors were attached to the boat with a bracket made of aluminum profile. Fig. 2. presents a picture of that. The rack proved to be well functional and adaptable. Data was gathered on five different boats.



Figure 2. Test equipment installed to a boat

The sealing of the device was tested before the actual dive so that any leaks would not cause equipment failure. The pressure hulls were tested by pressurizing them with water to a pressure of 5 bars. The operation of the thruster unit and engine controller were tested in a water tank to ensure that water did not affect the operation of the thrusters or cause short circuits. The motor cable connections were also tested at a water pressure of 5 bars.

V. RESULTS

In Fig. 3., the assembly of the vehicle was almost ready. The dive tests of the device have not yet begun, although many different areas have been tested. Testing has been done physically with tightness tests, testing of individual thrusters in a pool of water, software in data gathering, teaching neural networks, and running software sections in simulations. To date, no insurmountable obstacles have occurred, and it was expected that the device will also pull through the dive test. The prototype came to cost ca. 9.500€ where the sonar components are the most expensive components, ca. 4000€. Thrusters, thruster controllers and the computer cost ca. 1000€. Battery pack and BMS (Battery Management System) costs 1500€.

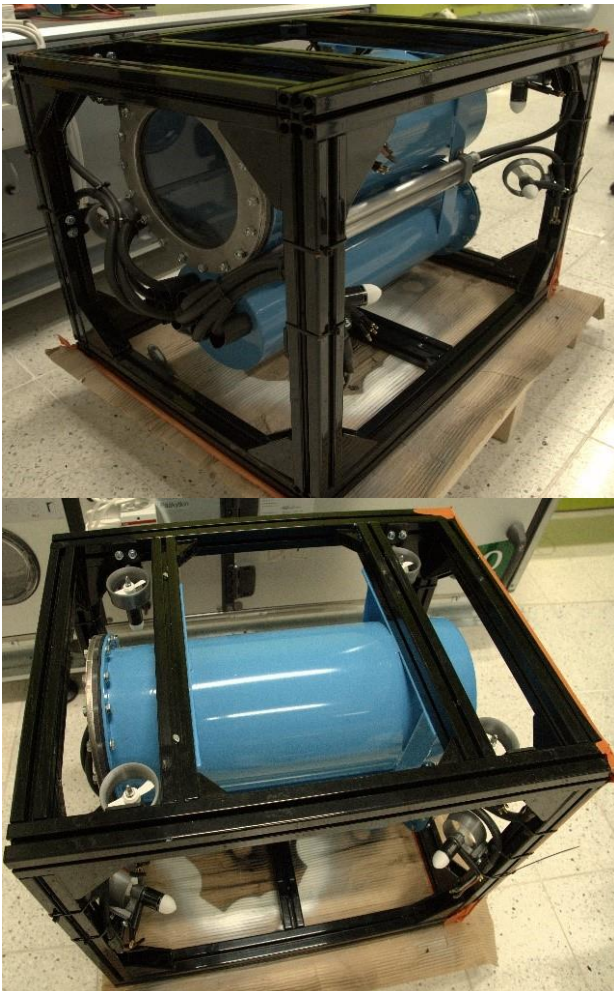


Figure 3. Pictures of the assembled vehicle

The first pressure test revealed that the seal type of the flanges was not very suitable for the purpose and was replaced with rubber. In addition, fastening bolts were added. After these modifications, the flanges were tight. The water tightness of the motor cable endings situated on the motor-side were also successfully tested at a pressure of 5 bars.

The thruster unit and motor controller perform well in the tests. More detailed thrust measurements have not yet been made, but the feeling is that there is enough thrust to move the device at the desired speed of a few kilometers per hour. The interconnection of the motor controller and the ROS software was also successfully tested.

The sonar was tested in data gathering. Data transfer with an HDMI-USB converter to a computer worked reliably, and no problems occurred after several hours of use. The associated ROS code was also tested at the same time. Optimal settings were sought for the sonar settings so that the image would be as good as possible in different ground conditions and that the targets would be identified as well as possible.

Preliminary results of the velocity calculated from the sonar image are encouraging. The velocity error compared to GPS speed is on average less than 0.2 km / h. With the increasing amount of data, the error is expected to decrease even more. It should also be noted that GPS speed is not accurate, especially at low speeds. In some situations where the measured GPS speed is incorrect, the calculated speed may be closer to the truth. How these results correlate to a practical situation where there are almost limitless new types of bottom shapes, remain to be seen.

VI. CHALLENGES

Obtaining location and velocity information below the surface is largely different from obtaining similar information in the air or on the ground, although there are some similarities. The biggest difference is the lack of GPS and similar global positioning systems. In the water radio signals do not travel deeper than the surface layers, so these cannot be used. DVL is the standard way to do velocity measurement in commercial more expensive underwater devices. The problem here is mainly the price, which limits its use in this implementation. With the selected sensors, the movement of one axis can be calculated quite accurately from the sonar image, and the pressure gives a second axis, but in the lateral direction one must rely mainly on the information of the impeller sensor and the accelerometer.

VII. CONCLUSION AND FUTURE STUDIES

This study presents the prototype of an autonomous underwater vehicle. The system was built using commonly available sensor components in combination with an open source code and in-house designed parts. The configuration of the autonomous underwater vehicle was introduced in this study. The results obtained from the component tests performed in the actual operating environment were presented. Based on these tests, the subsystems were found to be well-suited for underwater environment.

There are couple of possible actions that could be taken into consideration in the future research. Although, the Kalman filters have been a standard way to make sensor fusions for a long time, their ability to respond to changing circumstances are limited. As more data accumulates from the device tests, a neural network could be a better solution for this. It could be easier to teach the neural network, for example, situations where a certain sensor does not provide the correct information or does not provide information at all.

Another more researchable action could be developing a proper simulation model using different neural networks in Gazebo simulation environment. The simulation done by the SubjuGator project provides a good basis for the simulation model. To simulate the robot, the robot data as a URDF model is required. For the simulation, the device's water resistance information is needed, as well as the thrust produced by the motors at different power values. However, based on the data from the test run, they can be calculated. The most challenging will probably be the simulation of the sonars. The Gazebo does have LiDAR-type sensors modeled, so this could probably be a good starting point.

ACKNOWLEDGMENT

This work has been supported by Ulla Tuominen Foundation SR, High Technology Foundation of Satakunta, WinNova Länsirannikon Koulutus Oy Ltda and Satakunta Rescue Department Unit of Rauma, Finland. Helkama Bica

REFERENCES

- [1] J. Gelli, A. Meschini, N. Monni, A. Ridolfi, L. Marini, B. Allotta Development and Design of a Compact Autonomous Underwater Vehicle: Zeno AUV DOI: <https://doi.org/10.1016/j.ifacol.2018.09.463>, 2018
- [2] H.Niu, S. Adams, K. Lee, T. Husain, N. Bose Applications of Autonomous Underwater Vehicles in Offshore Petroleum Industry Environmental Effects Monitoring DOI: <https://doi.org/10.2118/09-05-12-GE>, 2009
- [3] H. R. Widditsch SPURV – The First Decade, University of Washington, Applied Physics Laboratory, 1973.
- [4] W. E. Nodland, T. E. Ewart, W. P. Bendiner, J.B. Miller, and E. E. Aagaard SPURV II - AN UNMANNED, FREE-SWIMMING, SUBMERSIBLE DEVELOPED FOR OCEANOGRAPHIC RESEARCH, University of Washington, Applied Physics Laboratory, 1981.
- [5] Autonomous Undersea Vehicle Applications Center. AUV System Spec Sheet. http://auvac.org/configurations/view/57?from_search=1, 2018.
- [6] Moline. M. A., Blackwell S. M., von Alt C., Allen B., Austin T., Case J., Forrester N., Goldsborough R., Purcell M. & Stoekey R. Remote Environmental Monitoring Units: An Autonomous Vehicle for Characterizing Coastal Environments. Journal of atmospheric and oceanic technology, 2005.
- [7] A. Mahtani, L. Sánchez, E. Fernández, A. Martínez, "Effective Robotics Programming with ROS", 3rd ed. Birmingham, Packt Publishing Ltd, 420 s. ISBN 978-1-78646-365-4, 2016.
- [8] UUV simulator, retrieved May 12, 2020, from <https://uuvsimulator.github.io/>
- [9] Robosub UCR, "SeadragonAUV", retrieved May 12, 2020, from <https://github.com/robosub-ucr/SeadragonAUV>
- [10] D. Volya, J. Brown, A. Puldio, N. Suhlman, E. M. Schwartz "SubjuGator 2019: Design and implementation of a Modular, High-Performance AUV, 2019
- [11] J. E. Naglak, B. R. Page, N. Mahmoudian, "Backseat Control of SandShark AUV using ROS on Raspberry Pi", DOI: [10.1109/OCEANS.2018.8604630](https://doi.org/10.1109/OCEANS.2018.8604630), 2018
- [12] M.Carreras, C. Candela, D. Ribas, N. Palomeras, L. Magí, A. Mallios, E. Vidal, É. Vida, P. Ridaio, "Testing SPARUS II AUV, an open platform for industrial, scientific and academic applications", <https://arxiv.org/abs/1811.03494>, 2018.
- [13] T. Seta, A. Okamoto, S. Inaba, M. Sasano, "A New ROS-Base Operation Software for an Autonomous Underwater Vehicle HOBALIN and Its Test in Real Sea", DOI: [10.1109/OCEANSKOB.2018.8559223](https://doi.org/10.1109/OCEANSKOB.2018.8559223), 2018
- [14] E. Gallimore, R. Stokey, E. Terrill, "Robot Operating System (ROS) on the REMUS AUV using RECON", DOI: [10.1109/AUV.2018.8729755](https://doi.org/10.1109/AUV.2018.8729755), 2018
- [15] S. Zavari, O. Usenius, T. Salomaa, J. Villa Escusol, A. Heininen, J. Laitinen, J. Aaltonen, K. T. Koskinen, "Mechatronic Architecture Development of UX-1", DOI: [10.23919/ICCAS.2017.8204483](https://doi.org/10.23919/ICCAS.2017.8204483), 2017.
- [16] S. Zavari, O. Usenius, J. Villa, A. Heininen, T. Salomaa, J. Laitinen, J. Aaltonen, K. T. Koskinen, "Propulsion system development and power consumption in an autonomous underwater vehicle", DOI: [10.1109/ICSTCC.2019.8885503](https://doi.org/10.1109/ICSTCC.2019.8885503), 2019.