

IMECE2021-69347

IMPLEMENTATION OF 2-DOF GRABBER ARM AND COMPUTER VISION ON REMOTELY OPERATED UNDERWATER VEHICLE

Pascal Spino
 Washington State University
 Pullman, WA, USA

Konstantin I. Matveev
 Washington State University
 Pullman, WA, USA

ABSTRACT

The development and applications of remotely operated and autonomous underwater vehicles have significantly increased in recent years. As these vehicles operate in the harsh underwater environment, demanding requirements for their design usually reflect in a high cost of underwater systems. However, with more readily available inexpensive electronics and powering systems, lower cost developments have been initiated. In the present work, several modifications of a low-cost remotely operated underwater platform are described. One is a construction of a two-degree-of-freedom arm for manipulating underwater objects. The second is an improvement of the propulsion control on the vehicle to allow for gradual variation of thrust forces instead of the original on/off mode. The third enhancement is a computer vision system for identifying underwater objects of interest that is applied for automated steering of the vehicle. Initial tests with these elements in a laboratory tank are presented and discussed. They include (1) autonomous detection of a target and maneuvering towards it, (2) grabbing and moving an object with manual remote control, and (3) the combined test with autonomous identification, grabbing, and moving of a target. The reported developments and test results can help other researchers pursuing low-cost developments of underwater vehicles.

AUV	Autonomous underwater vehicle
DOF	Degree of freedom
CV	Computer vision
PD	Proportional-derivative
ROV	Remotely operated vehicle
PWM	Pulse width modulation

1. INTRODUCTION

The deployment of remotely operated vehicles (ROV) and autonomous underwater vehicles (AUV) has drastically expanded in recent years due to technological advances in electronics, structures, and compact energy sources. They are used for a variety of tasks, including collection of ocean data, inspection of marine structure, performing naval missions, and other assignments [1,2]. The market for ROV and AUV is predicted to continuously and rapidly grow in the near future [3].

As these vehicles operate in the water environment under elevated pressure and limited options for communication, demanding requirements to their design and manufacturing usually reflect in high costs of underwater systems. However, with proliferation of relatively inexpensive electronics and powering systems suitable for small craft, as well as advanced manufacturing techniques, a number of low-cost developments in this area have been initiated [4-6]. The present work belongs to this field, building up on our previous experience with marine and amphibious vehicles [7,8]. The developed platform presents a highly modifiable, multi-capable package at an expense significantly below conventional ROV / AUV. Constructed from less than \$500 USD in parts, this vehicle is about ten times less costly than LoCO-AUV [9] or BlueROV2 [10], which are also considered to be relatively inexpensive platforms.

The starting point in this project was a basic platform [11] consisting of a plastic-tube frame, three thrusters, and a video camera (Fig. 1), connected to control and powering modules above the water. Two propulsors provide differential thrust for

NOMENCLATURE

d	Gripper opening width
F_i	Force component along the i -axis
k	Constant coefficient
L	Arm extension length
M_i	Torque component along the i -axis
t	Time
T_t	Thrust force by the 't' thruster
$\mathbf{v}_s$	Control vector for subsystem 's'
T_{o1}	Translation vector
R_{ij}	Rotation matrix

horizontal-plane movements and steering, while the third, vertically oriented propulsor is used for diving. The thrusters initially operated only in simple on-off modes. As functionality and navigation capabilities are very important characteristics for underwater vehicles, several modifications on this platform have been implemented, including a computer vision system, more sophisticated thruster control, and a manipulator arm. They are described in the next section. Several tests performed with the enhanced system are presented afterwards. While other papers on the subject provide brief design overviews, this work aims for a comprehensive description in sufficient detail to replicate.

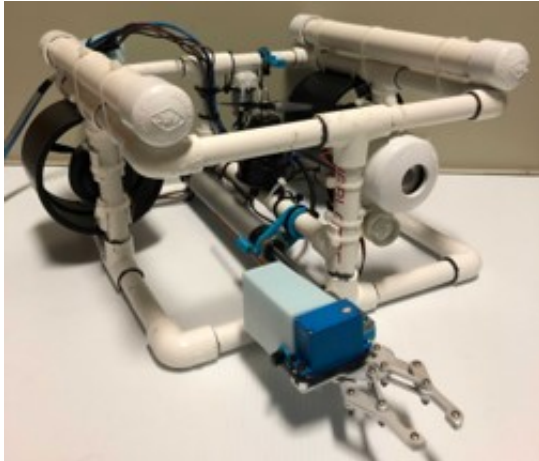


Fig. 1 ROV platform with implemented modifications.

2 EXPERIMENTAL SETUP

2.1 Base Hardware

The ROV presented by this paper is developed from the ROVIAB kit by Inventivity [11]. The frame, thrusters, camera, and some elements of the electrical system were retained in the present design. These components constitute the ‘Base Hardware’ of the ROV. This system, together with the added manipulator arm, is shown in Fig. 1.

The frame is primarily developed from 1/2" PVC pipe segments and 1/2" PVC pipe fittings. All frame sections are fitted with PVC sealant to prevent flooding during submersion for buoyancy purposes. The two forward thrusters are mounted to the frame with modified PVC tee's and are protected by a 4" ABS prop guard. The single dive thruster is mounted along a pipe segment near the buoyancy and gravity centers of the frame. Each thruster, of which there are three, is a 500 GPH, 12V bilge pump with the impeller being replaced by 4x4.5 propeller. The underwater camera is a 12V RCA compatible camera module sealed in clear epoxy for waterproofing purposes. The underwater light is similarly a 12V LED light sealed in clear epoxy. The power supply is a 12V, 7Ah SLA battery that remains out of water, supplying power to the ROV through a tether. The tether itself consists of three CAT5 cable lines that distribute power to each of the thrusters, camera, and light while receiving RCA signal from the camera. The original kit featured manual

switches for controlling each thruster, but these were removed in this project.

2.2 Manipulator Arm

The ROV has been modified with a custom-built two-degree-of-freedom manipulator arm (Fig. 2). This novel design belongs to a field of underwater manipulators and marks the lowest-cost construction found in academic work. It has the capability of linear forward/reverse motion up to 10 cm, and the end effector is capable of gripping objects up to 5.4 cm in diameter. The arm is rigidly mounted within the frame of the ROV for stability purposes and extends outwards at a maximum rate of 3.3 mm/s for environmental interactions. A block of foam is mounted to create a buoyancy volume at the end effector (Fig. 2b), allowing for pitch and roll stability throughout the full range of arm expansion. Once an object has been gripped, the arm can be retracted back into the frame to minimize shifting of the ROV's gravity center. Heavy objects can still unfavorably shift the gravity center due to the light-weight nature of the overall unit.

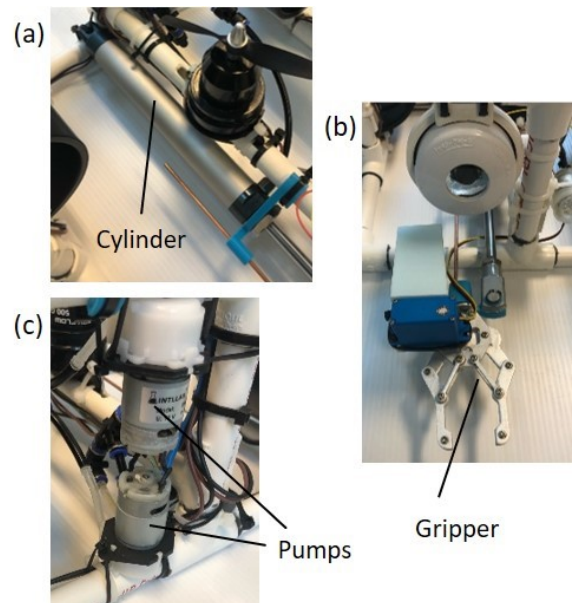


Fig. 2 Photographs of constructed manipulator arm elements: (a) extending cylinder, (b) end effector, (c) peristaltic pumps. Reproducible for less than \$100 USD.

The extending portion of the manipulator arm is a 25-mm bore, 100-mm stroke double action cylinder filled with water (Fig. 2a). This cylinder is driven indirectly by two 12V peristaltic pumps (Fig. 2c), which run in parallel to drive water in a closed cycle to and from the cylinder chambers. The two pumps can operate at variable speed for variable extension speeds. The end effector is mounted directly to the connector of the air cylinder extension rod. An additional 3.5-mm-diameter rod is also attached to the effector that extends with the cylinder to restrict any rotation while the cylinder expands.

The end effector is a parallel jaw gripper, driven by a servo motor through a system of linkages (Fig. 2b). Unlike the brushed motors of the peristaltic pumps, a standard servo motor is not suitable for aquatic environments without waterproofing. The gripper is driven by an HS-646WP waterproof servo from Blue Robotics, operating at 7V. This servo is capable of up to 90° rotation, which is ample for full range of motion on the gripper. Care is taken in the tether arrangement to shield the control signal wire from thruster cables due to interference issues while the thrusters are operating.

2.3 Control System

The control system is custom-developed and consists of Arduino-based hardware as well as programs in C++ and Python. This system facilitates serial communication for both thruster and manipulator control. The modular design presents a simple interface for full ROV control via USB connection by an external computer, which is not a conventional approach that also serves to reduce cost.

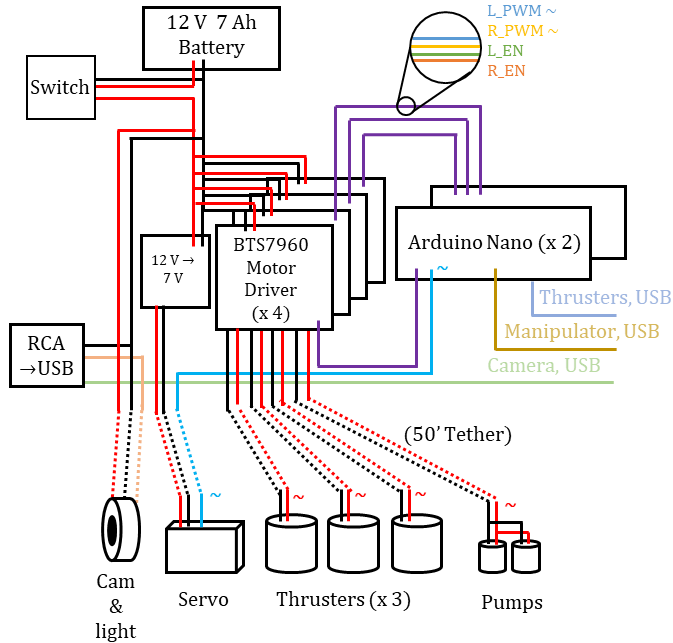


Fig. 3 Schematic of employed hardware.

Each of the ROV's three thrusters and manipular arm pumps employ brushed 12V motors. Variable speed control is accomplished via Pulse Width Modulation (PWM), by which the duty cycle of supplied voltage is controlled to emulate a variable voltage signal [12]. Four BTS7960 motor drivers manage PWM signal to each of the motors, receiving 12V from the battery and 0-5 V from the control boards to produce a 0-12 V PWM motor supply (Fig. 3). There are two control boards in total, both of which are an Arduino Nano. One board manages hardware control for the three thrusters, and the other board manages all elements of the manipulator arm. All non-waterproof electrical hardware is housed in a container outside water, passing control

signals to the submerged ROV hardware via the tether. While this architecture results in a thicker tether than for similar vehicles, the cost of production, reconfiguration barriers, and vehicle size is reduced. Each Nano executes a script developed from custom C++ libraries that continuously converts strings of desired control vectors to corresponding electrical signals.

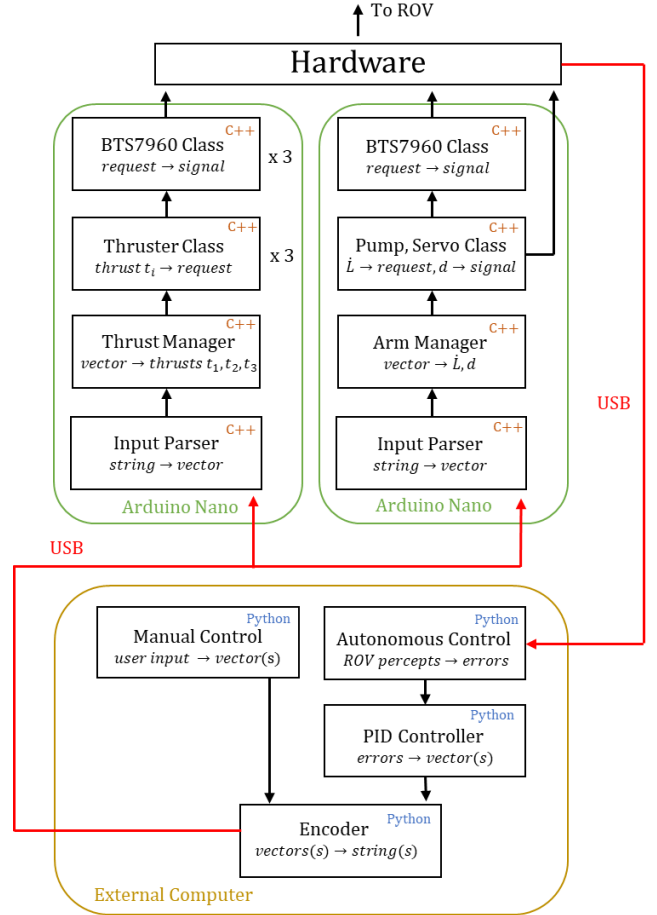


Fig. 4 Arrangement of programs on computing boards.

The C++ control codebase operates on multiple abstraction levels to transform desired control vectors to real-world equivalents (Fig. 4). Four pins are required to generate PWM signals in each motor BTS7960 drivers. The low-level operation of these pins by the Arduino Nano is abstracted to percent thrust requests through a series of C++ objects. In the highest-level class, vector requests for overall thrust and manipulator movement are decomposed to individual component requests. In the case of thruster control, the thrust state of the ROV is represented as a three-dimensional vector with two translational and one rotational component, as shown in Fig. 5a. The vector decomposition to individual thrusts for this thruster configuration is straightforward. The manipulator state vector is represented as a two-dimensional vector of the arm extension force and inter-jaw distance of the gripper, which is also decomposed to pump force and servomotor rotation (Fig. 5b).

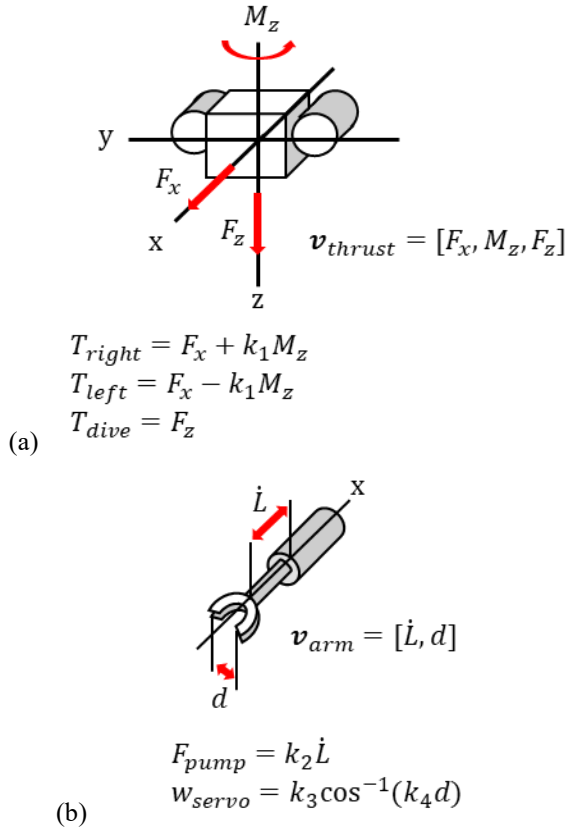


Fig. 5 Schematic representations of actuating elements and control commands for (a) ROV motion and (b) end effector.

Each of the thruster and manipulator control objects has a defined *vectored()* function that continuously reads vector-representational strings from serial communication. The strings also contain an additional checksum element that mitigates communication errors, disabling all thrusters if the checksum requirement is not met. These string commands are all that is needed to operate the ROV and can be sent to the Arduino Nanos via USB or I2C pins by any method of serial communication. The pySerial module was selected in conjunction with Python3 scripts for the manual and autonomous control discussed later.

2.4 Autonomy

Two autonomous tasks performed by the ROV are presented in this paper. In both cases, the only sensor was an onboard camera. The Open Source Computer Vision library (OpenCV) was used for object detection and fiducial marker recognition.

All autonomy algorithms were implemented on an external computer that had access to both the ROV camera and control system. The camera frame data was collected as RCA signal, transported along CAT5 tether cable to a USB-converter in the control case, and read into a python script for processing. General object detection, as demonstrated by navigation to the bottle-type object in Section 3, utilized the OpenCV Deep Neural Network model with training from the COCO dataset [13]. With

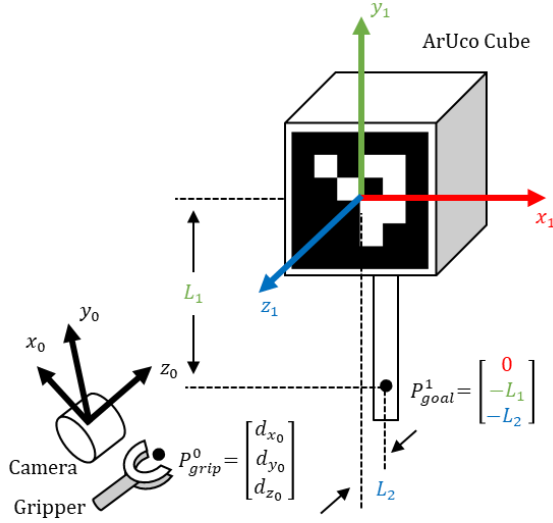
this model, the ROV could recognize common objects such as a dish soap bottle (used in one of the tests) in terms of rectangular bounding boxes and confidence intervals. While accurate above water, optical effects lead to appreciably more noise and misidentifications once submerged. To combat this, identifications with less than 20% confidence were discarded and estimated object locations were averaged over three successful identifications rather than frame-by-frame. Once the object bounding box was consistently captured, a simple state machine was implemented for the ROV to (i) search for the bottle, (ii) approach the bottle to a certain distance, and (iii) retreat from the bottle. For the latter two states, thrust vector commands were created by a PD controller that estimated the positional pose of the bottle from XY frame locations and size in order to keep it centered to a given distance.

The next development was to expand this autonomy task to object manipulation. To improve detection accuracy, a special cubical object was devised with fiducial markers labeling each face. Specifically, the ArUco fiducial marker was selected due to support in OpenCV [14]. A fiducial marker allows for full pose estimate as a uniquely oriented binary matrix. By calibrating the camera to account for distortion effects and providing a known size of the marker, the OpenCV *detectMarkers()* function can calculate the 3D position and orientation of an ArUco marker with reference to the camera coordinate system. Camera calibration is performed by computing the intrinsic camera matrix and distortion coefficients, which can be accomplished by capturing several checkerboard images with the camera and utilizing OpenCV calibration functions. To correctly navigate the gripper to the gripping goal, coordinate transformations were performed on detections of the cube face(s) to describe a vector from the gripper to the grip-location of the cube (Fig. 6). Given a rotation matrix and translation vector that described the transformation between any cube face and the camera coordinate system, this vector can be computed with knowledge of the cube's grip location and the location of the gripper with respect to the camera frame. It is possible that the ROV detects 2 cube faces at once, in which case the vector is averaged over each computation.

The ROV operated on a state machine to (i) search for the cube, (ii) approach and grip the cube, and (iii) bring the cube to a new location within the tank. Additional states were implemented to verify the gripping process and repeat the approach if unsuccessful (Fig. 7). The ROV could verify a grip success by computing the same gripper-to-grip-location vector shortly after executing a grab. If this vector was unreasonably large, or no detection was made, the grip was considered unsuccessful.

All autonomy processes involving vision were reliant on the PD controller. A PD controller is a common closed-loop control mechanism that maps an error term to a measured process value, where both error proportion and error rate of change are considered. In this case, the PD controller converts pose error estimates between the ROV and its' goal location into consistent thrust vectors for approaching that location. This controller was implemented in Python to separately control each of forward,

turning, and diving motions by the ROV. The proportional and derivative constant terms for each motion were described in two intervals, such that the ROV made aggressive approaches at large distances and much finer approaches when close to its' target. The controller constants were manually tuned through experimentation.



Given $T_{01} \in \mathbb{R}^{3 \times 1}$, $R_{01} \in \mathbb{R}^{3 \times 3}$ from Camera Detection:

$$v_{grip,goal} = R_{01}P_{goal}^1 + T_{01} - P_{grip}^0$$

Grip when $v_{grip,goal} \approx \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

Fig. 6 Representation of computer vision-based goal for the gripper position.

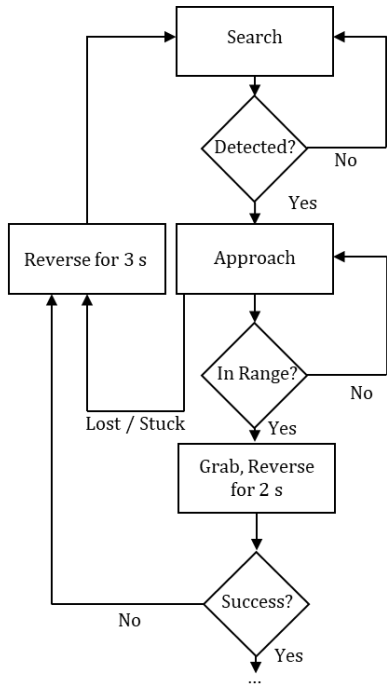


Fig. 7 Algorithm for search-approach-grab.

2.5 Pose Data Collection

The ROV did not have any onboard positional sensors or odometry capabilities. For data collection purposes, another computer vision system was devised with use of OpenCV for recording the ROV pose during experiments. This system consisted of an overhead camera for the water tank and an ArUco marker attachment module for the ROV (Fig. 8). The overhead camera was suspended such that it had a full, unobstructed view of the tank. The ArUco module was positively buoyant and attached to the ROV with two sliding rods such that it floated on the water surface when the ROV submerged. This allowed the marker to remain in view of the camera and minimally affect the ROV's balance when diving. Such a system has not been found in related literature, presenting a novel approach to gather accurate positional data with minimal alteration to the underwater vehicle itself.

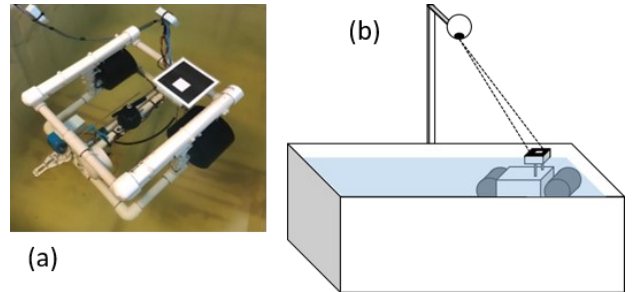


Fig. 8 Computer vision system for recording ROV pose.

Accompanying python code was developed with usage of OpenCV to derive the ROV pose from detections of the marker. To improve performance, the program only recorded and stored raw images from the camera in real time and processed the images afterwards. Processing involved undistorting the image, computing the pose of the ArUco marker, applying transformations to compute the location of the ROV's center, and outputting this data as a time series.

3. LABORATORY TEST RESULTS

Several exploratory tests with the modified ROV have been conducted in a laboratory water tank. The first was on validating that the CV-based motion control allows the vehicle to approach a target. A sample image sequence obtained in this experiment is given in Fig. 9. The target was a bottle with blue liquid attached to a wall opposite to the initial ROV position, while the ROV was initially oriented away from the target ($t = 0$ s in Fig. 9).

In the beginning the ROV scans the space in search of the bottle, for which it has been pre-trained to recognize with a convolutional neural network. Once the ROV detects the target ($t = 4$ s in Fig. 9), the vehicle starts moving towards it by engaging both horizontal and diving thrusters to bring the target into the center of the ROV camera view while approaching to a distance of about 20 cm between the camera and the bottle. The vehicle reaches the desired proximity of the target at $t = 15$ s (Fig. 9), and retreats after that.

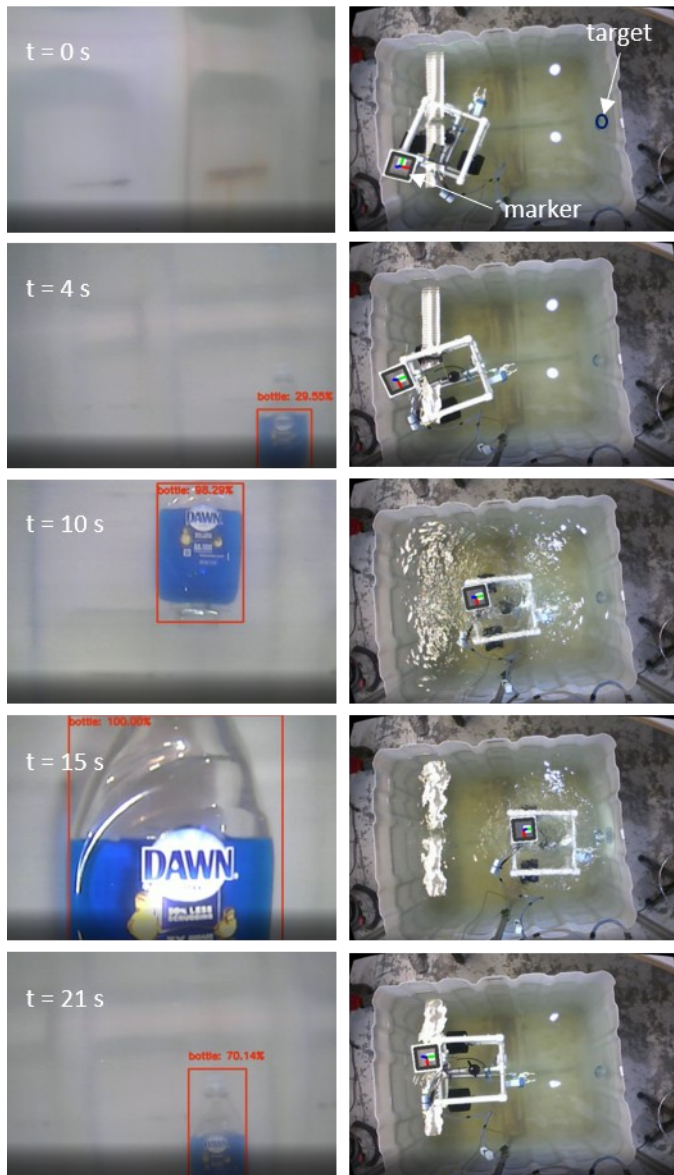


Fig. 9 Sequence of images from ROV camera (left column) and top camera (right column) in the target-approaching, computer-vision test. Red box indicates recognized bottle image, and a number above the box is the confidence level in identifying the correct object.

To record the ROV horizontal position and experiment with the computer vision system for another process, an overhead camera was used to detect the ROV position and yaw angle using a marker attached to the vehicle (as described in section 2.5). ROV locations recorded this way and target positions in the vehicle camera view are shown for the test duration in Fig. 10. One can notice oscillatory motions indicating the PD controller has not been optimized, although the target was reached successfully.

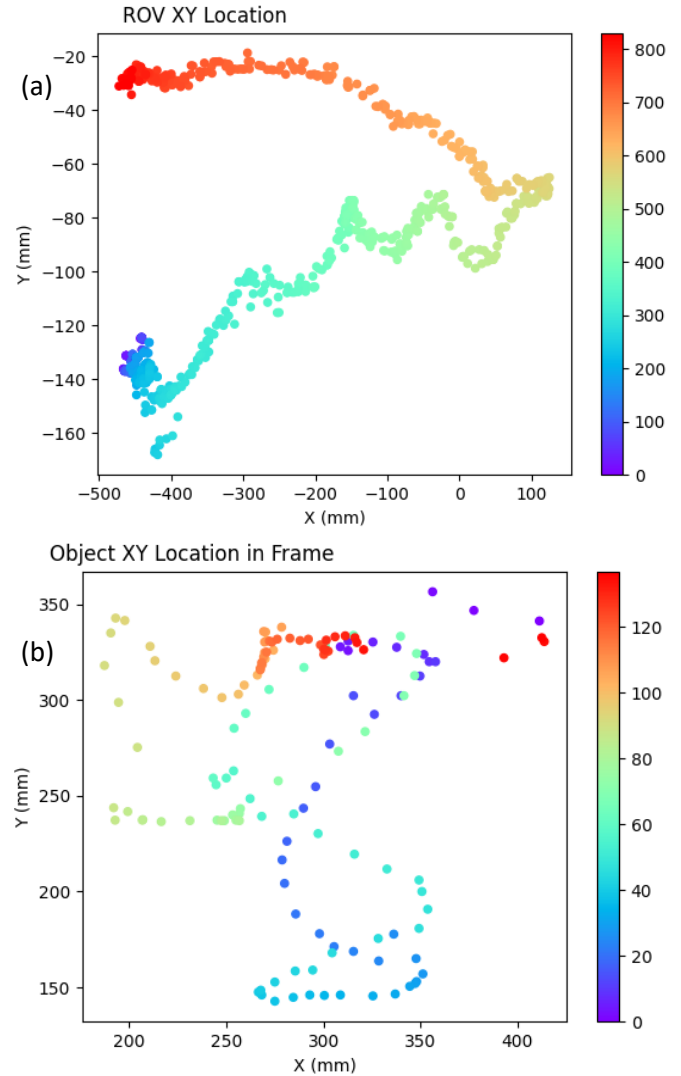


Fig. 10 (a) Position of ROV measured with top camera and (b) object location in the ROV view in the target-approaching, computer-vision test. Color bars show the time step numbers ($\Delta t \approx 0.04$ s and 0.2 s for the top and ROV cameras, respectively; y-position for ROV view is in horizontal direction).

The time history of the command signals sent to the ROV thrusters by the controller and the vehicle's yaw angle are shown in Fig. 11. The yaw variable is given with about 6 s lag in this figure, as the top and ROV cameras started at a different time. Only the ROV camera was used for the vehicle control, while the top camera simply recorded the vehicle pose. One can see the initial constant signals going to horizontal thrusters (2-5 s in Fig. 11a), while yaw changes almost linearly (8-11 s in Fig. 11b). After detecting a target, all three thrusters become engaged to move towards the target. Again, oscillatory motions in both thrust commands and yaw angle are noticeable. Upon reaching the desired point, thruster signals become negative indicating backward motion away from the target. This test validated the deployment of the computer vision-driving mode.

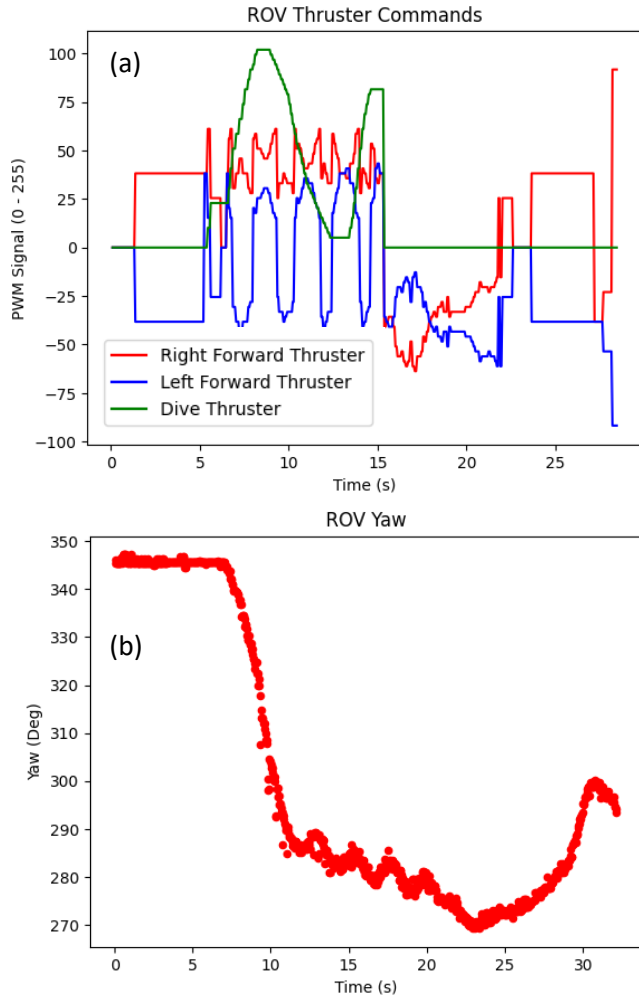


Fig. 11 (a) Command signals sent to ROV thrusters and (b) yaw angle of ROV measured with the top camera.

Another test was conducted on operation of the manipulator arm under operator control from a connected laptop. A rubber puzzle ball suspended underwater served as an object of interest. The ROV horizontal positions and yaw angle are captured by processing video from the overhead camera (Fig. 12a,b). The command signals sent to horizontal thrusters are given in Fig. 12c. A sequence of images from the top camera in the recorded test is illustrated in Fig. 13.

The ROV is first directed towards the target and an arm is used to grab the object. Then, the ROV is moved to another part of the tank, the object is released, and the vehicle retreats. Initially, short actions are taken by the operator to approach the target; after that, longer commands are exerted to quickly move the object to a different location. The dive thruster was not utilized in this test since the object was located at a depth similar to that of the grabbing arm. Views from an ROV camera of the ball grabbed by the arm captured in another similar test are shown in Fig. 14. Thus, the functionality of the arm was confirmed.

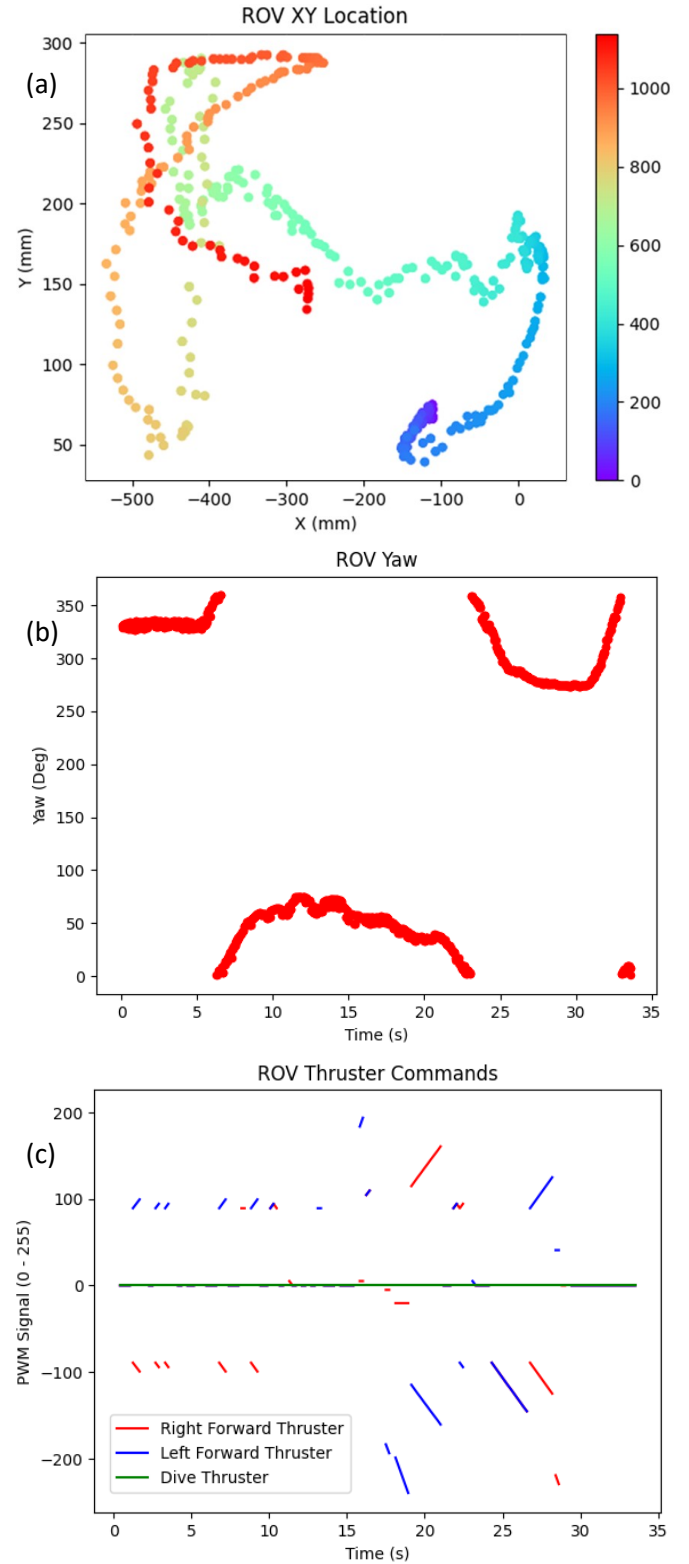


Fig. 12 (a) Horizontal position of ROV measured with top camera, (b) yaw angle, and (c) command signals sent to thrusters. Color bar in (a) show the time step numbers ($\Delta t \approx 0.03$ s).

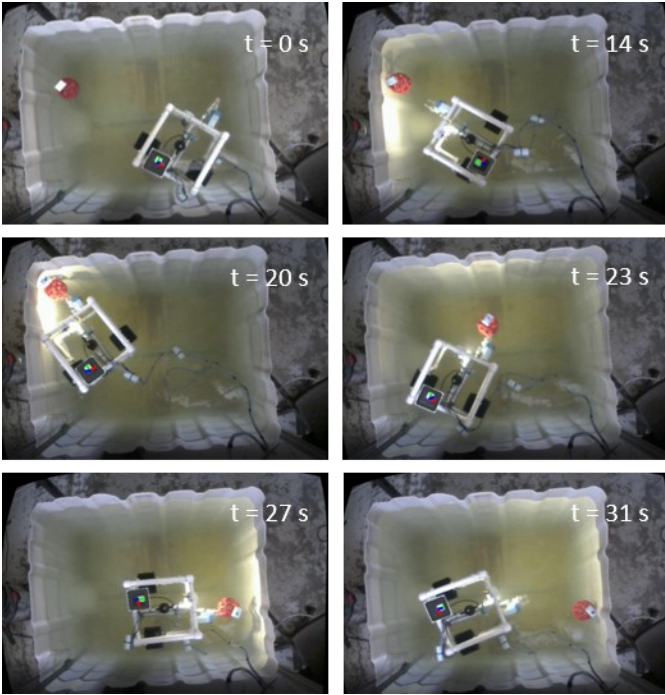


Fig. 13 Sequence of images test from the top camera in the test of grabbing and moving an object.

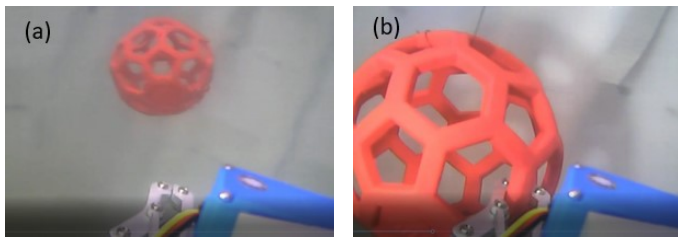


Fig. 14 Images from ROV camera: (a) approaching target, (b) grabbing target.

After experimentally verifying that the implemented functionalities (manipulator arm and vision-based motion) work reasonably well in the present setup, the next development goal was to combine them and demonstrate that the vehicle can autonomously search for and approach the target, grab it, and bring to another location. Since capturing the object in the autonomous mode requires precise relative positioning, markers were added to the target in the present combined test to reduce error in determining the relative position.

Even with this modification, there is a possibility that the vehicle may fail to capture the object due to disturbances and small size of the target's component that the gripper intends to grab. Therefore, the autonomous control algorithm allowed the vehicle to recognize failures and continue searching attempts if the object was not securely captured.

The information presented below is from a test in which the vehicle needed three attempts to secure the target (although only

one attempt was needed in other cases). The images showing the vehicle position in the tank and views from the onboard camera are illustrated for several times moments in Fig. 15. The vehicle trajectory and thrust command signals for the entire test duration are given in Fig. 16, while the relative distances to the target are shown separately for each of the three attempts in Fig. 17.

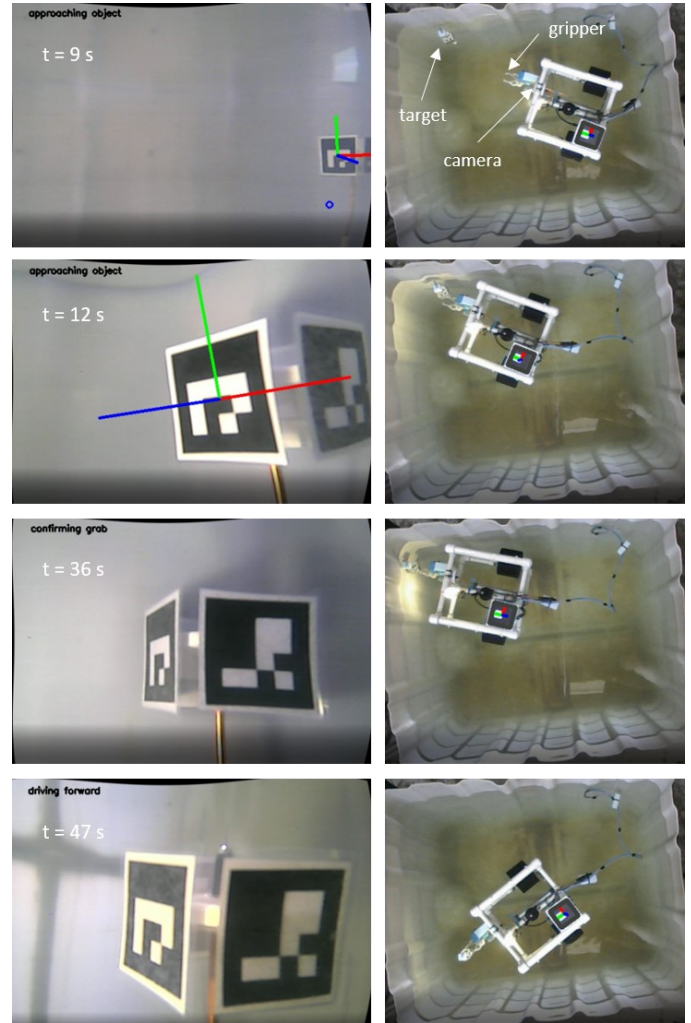


Fig. 15 Sequence of images from ROV camera (left column) and top camera (right column) in the autonomous detection/grabbing test: 9 s, first target detection; 12 s miss in the first approach; 36 s, capturing in the third attempt; 47 s, releasing in different location.

The ROV first scans the area by rotating itself (with opposite signals to thrusters in Fig. 16b). Once the target is detected at 9 s (Fig. 15), the vehicle approaches (with similar and receding in time thrust signals in Fig. 16b). As the first attempt is a miss (Fig. 15), the ROV moves back (Fig. 16). Only in the third attempt at 36 s, the target is captured (Fig. 15). After this, the vehicle retreats and moves to another corner of the tank where the object is released (Figs. 15,16). In the relative distance plots (Fig. 17), one can notice that the ROV spends some time in the vicinity of

the target trying to perform and validate a grab in the first two attempts, while in the last successful attempt it quickly reaches and grabs the object. In this test, the combined functionality of the vision-based motion and the effector was verified.

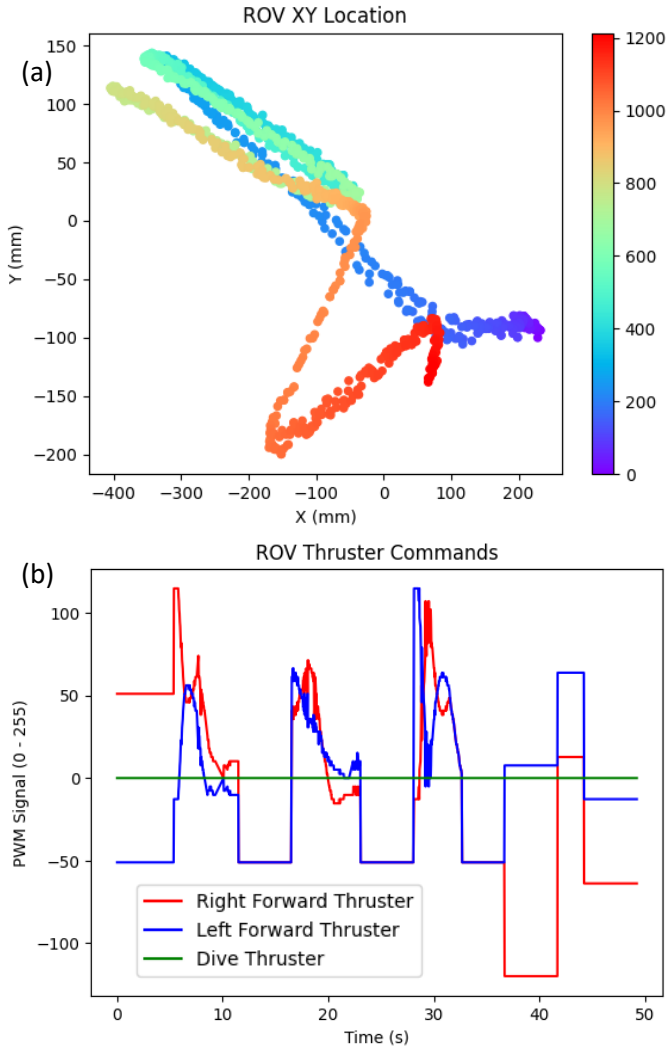


Fig. 16 (a) Position of ROV measured with top camera and (a) Command signals sent to ROV thrusters ($\Delta t \approx 0.04$ s).

4. CONCLUDING REMARKS

A basic, low-cost remotely operated underwater platform was augmented in this work with gradual motor control, a 2-DOF grabbing arm, and a computer vision-based motion control system. Tests conducted in a laboratory tank confirmed the enhanced functionalities of this ROV. Possible future development steps include moving all control electronics and energy sources inside a watertight container on the vehicle to provide untethered capability for this platform, testing it in open-water reservoirs, and instrumenting with additional sensors, effectors and communication modules for broadening its potential practical applications.

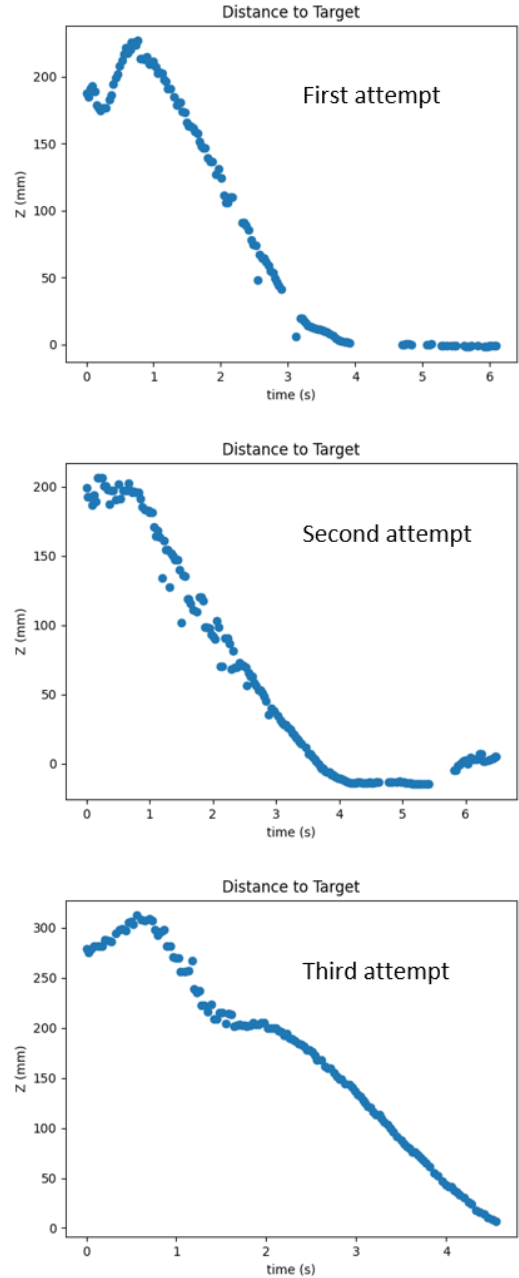


Fig. 17 Distances from the gripper to the target in three attempts.

ACKNOWLEDGEMENTS

This material is based upon work supported in part by the by the National Science Foundation under Grant No. 1800135.

REFERENCES

- [1] Petillot, Y.R., Antonelli, G., Casalino, G., Ferreira, F., 2019, Underwater robots, IEEE Robotics and Automation Magazine, June 2019, 94-101.

[2] McLean, D.L., Parsons, M.J.G., Gates, A.R., Benfield, M.C., et al., 2020, Enhancing the scientific values of industry remotely operated vehicles (ROV) in our oceans, *Frontiers in Marine Science*, April 2020, 7, 200.

[3] www.techsciresearch.com/report/global-offshore-auv-rov-market/3336.html

[4] Vuksic, M., Josipovic, S., Coric, A., Kraljevic, A., 2017, Underwater ROV as inspection and development platform, *Transactions on Maritime Sciences*, 6(1), 48-54.

[5] Tuan, P.V., Shpektorov, A.G., 2019, Software and hardware complex for the management of small-sized underwater vehicle, *Procedia Computer Science*, 150, 147-153.

[6] Duecker, D.A., Bauschmann, N., Hansen, T., Kreuzer, E., et al., 2020, HippoCampusX – a hydrobatic open-source micro AUV for Confined environments, *IEEE/OES Autonomous Underwater Vehicles Symposium*, St. Johns, NL, Canada.

[7] Matveev, K.I., 2008, Static thrust recovery of PAR craft on solid surfaces, *Journal of Fluids and Structures*, 24, 920-926.

[8] Thompson, N., Whitworth, P.R., Matveev, K.I., 2021, Development of small-scale unmanned hydrofoil boats, *Journal of Unmanned Vehicle Systems*, 9, 21-32.

[9] <https://loco-auv.github.io/>

[10] <https://bluerobotics.com/store/rov/bluerov2/>

[11] www.nventivity.com

[12] en.wikipedia.org/wiki/Pulse-width_modulation

[13] arxiv.org/abs/1405.0312

[14] docs.opencv.org/master/d5/dae/tutorial_aruco_detection.html