Small Surface Vessel for Multi-Robot Systems Education

Dino Dubinović, Leonardo Roy Sabolić, Nikola Borzić, Đula Nađ
Laboratory for underwater systems and technologies (LABUST), University of Zagreb Faculty of electrical engineering and computing (UNIZG-FER), Zagreb, Croatia
dino.dubinovic@fer.hr

***Abstract* — The use of robots in education is not new, and a plethora of mobile (land) based robots (and some aerial) exist to cover needs from primary to higher education. However, marine robots are not as prevalent due to the logistics of having adequate pool facilities and inherent challenges of aquatic environments. Nevertheless, the environmental challenges can provide students with a richer learning experience. Similarly, with marine tools to support education, students are exposed to the blue economy sector, which is a growing domain also from a robotics perspective.**

This technical paper documents the creation of a small unmanned surface vessel developed by students for students as part of an undergraduate project. The platform is designed to be scalable to multiple instances with the aim of supporting project-based teaching in a multi-robot systems course at the University of Zagreb Faculty of Electrical Engineering and Computing. As a collaborative effort between students and their mentor, the paper aims to highlight the technical and educational motivations for the creation of these vessels, followed by a technical overview of the electronics and supporting software.

Keywords – unmanned surface vehicle, marine systems, underwater IoT, multi-robot systems.

I. INTRODUCTION

In the past decades, there has been a trend towards including hands-on and project-based approaches to enhance the learning of complex topics. Methods like project-based learning structure the learning around an inquiry that students aim to answer through a guided project [1]. The frontal instruction method is reduced in favour of students proposing solutions (with arguments) to the instructor for encountered challenges. In the context of multi-robot systems, a simple inquiry might be in the direction of how to emulate flocking behaviour using multiple homogeneous robots. However, such a question implicitly assumes that a) the robots exist and are approachable to the students, b) conditions for robot operation (e.g., a larger area) are available, etc. Often, these assumptions are only partially satisfied and can cause students to spend more time handling shortcomings of the setup rather than focusing on acquiring the intended knowledge.

This technical paper is the initial stride in crafting an end-to-end solution for students in the multi-robot system class. The robot in this case is a small unmanned surface vehicle (USV) equipped with all the necessary software support to determine its position and be controlled remotely through a Robot Operating System (ROS) framework. What makes this platform intriguing is that it is envisioned to be created by students for students. Specifically, the development of this platform is an undergraduate student's project, with more advanced stages concentrated as a BSc thesis. The application of this platform will be in a graduate class on multi-robot systems.

The idea of using unmanned surface vehicles in education is not new, and some examples from the literature are provided in the background chapter. However, there are very few papers focused on documenting a USV for educational purposes that fits the operation of multiple units within a small pool, i.e., of a size not exceeding a length of 25cm. This is where we see the first technical contribution of this paper. The second contribution is the focus on selecting minimal footprint commercial off-the-shelf (COTS) hardware, allowing for further system reduction (length less than 15cm), while still providing sensor options for use in different projects.

In the next section, the background and related works are presented. This is followed by a description of the ideal system model and the constraints that defined the final implementation. Results are provided in the third section, along with a brief discussion on the observed shortcomings and future improvements.

II. BACKGROUND

The proposed unmanned surface vehicle is designed as an undergraduate project by the authors as part of the Laboratory for Underwater Systems and Technologies (LABUST). The laboratory has available pool facilities, shown in Fig. 1, currently utilized mostly for research projects, but the general trend at LABUST is towards maximizing the pool usage for students' classes at the University of Zagreb Faculty of Electrical Engineering and Computing.

Figure 1: The LABUST pool facilities [3].

Surface platforms, such as H2OmniX shown in Fig. 2, were developed by LABUST students before [2]. These platforms were used in various projects and eventually commercialized through a spin-off company. However, due to a lack of facilities up to two years ago, they were not utilized for teaching classes. Currently, the H2OmniX surface platform is used for classes, but due to its size, it is not suitable for cooperative or swarm-based scenarios in the available pool facilities.

Figure 2: The current commercial omni-direction vehicle H2OmniX that was initially developed by LABUST students and refined over almost a decade through different projects.

Currently, Sphero robots are used for projects in multi-robot systems. They can be controlled from ROS, but their operation requires an arena setup with multiple ceiling cameras for a 10-15 m2 area. The arena occupies lecture room space, so the downside is that the setup needs to be repeated after lecture room usage. On the other hand, the pool infrastructure already has camera facilities and a dedicated 40 m2 of water surface, avoiding the need to assemble and disassemble the arena [3].

Small, approachable commercial robotic systems already exist but are often focused on land applications. In recent years, aerial platforms have become available (e.g., Crazyflie). Sadly, there is little focus on small USVs that are needed for pool operation. One fine example of mini USVs targeted for educational use is described in [4]. The authors present a complete solution for building a 23cm differentially operated vessel. The single vessel price is marked above \$350, with mechanical parts (hull, shaft, epoxy, etc.) accounting for 50% of costs. Although not designed directly for the pool, the SmartBoat described in [5] is an affordable differential surface vessel coming in at a much lower cost than the previous example. However, the circular shape with a 28 cm diameter was too large for our requirements.

Apart from papers, there are low-cost solutions across the web with detailed build instructions (e.g., 3D printed RC jet-boat[1]). This example solution was tested and evaluated before the final version and presented problems with water-tightening. The main reasons were related to 3D printing performance and inexperience with waterproofing control shafts. The project focus was more on the electronics and software setup than hull design. This motivated looking into alternatives for procuring an RC boat hull. An RC boat[2] hull was found for €15 which already contained a differential thruster and watertight easy access hatch.

III. SYSTEM MODEL

The ideal model would be an omnidirectional surface vessel sized 5-10cm with complete position and attitude estimation. However, considering the cost and time constraints, the specifications were adjusted to be more realistic. The targeted size was slender hulls less than 25cm in length that could carry a single-cell Li-ion battery, thrusters, and a control board with MEMS gyro capable of RF communication. Initial specifications assumed that localization could be done using a few LEDs mounted on the hull, but upon revisiting the hull size, localization using markers was shown to be robust enough.

The RC boat used as the base is already equipped with two brushed DC motors. The simple remote controller that comes with the boat only controls the two motors in an ON/OFF fashion. While part of the targeted applications of the boat could function with binary thruster control, most brushed DC controllers already come with pulse-width modulation (PWM) capabilities allowing for more fine-grained control of thrusters. Therefore, for controlling the speed and direction of the motor speed, the DRV8833 dual-channel H-bridge was chosen. Its advantage lies in operational compatibility with Li-ion low-voltage, high-current capability, and ease of integration with Arduino and similar boards.

The Arduino Nano 33 IoT was selected as the main processor board. The advantage of the Nano 33 IoT is that it has integrated WiFi and BLE communication support as well as a 3-axial accelerometer and gyro. The drawback was that operation on a single-cell Li-ion is not safe and therefore a voltage regulator was added as one extra component to the system to ensure 3.3V across the complete battery operating range. The RC boat already comes with a 500mAh battery, which is good for 15-20 minutes of operation depending on thruster usage. However, since the targeted operation is towards 2 hours, the battery is upgraded to an 18650 3000 mAh cell. The electronics and PCB layout are shown in Fig. 3 and Fig. 4.

The Arduino software encompasses the control of the thrusters, thruster allocation, acquisition of the inertial sensors, and communication towards the poolside master computer. Additionally, tracking software was developed utilizing ceiling cameras above the pool to capture the marker for boat position and horizontal orientation. Data from the cameras is exchanged via UDP between the image processing computer and the Arduino Nano 33 IoT where it is fused with the onboard inertial measurements for a final filtered position estimate.

---

[1] https://www.youtube.com/watch?v=0LeV_MzVAqY

[2] https://www.conrad.hr/hr/p/rc-motorni-camac-za-pocetnike-rtr-245-mm-2384954.html

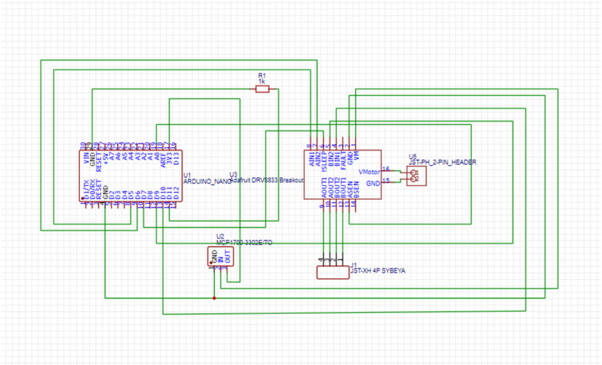

Figure 3: The electronics schematics.

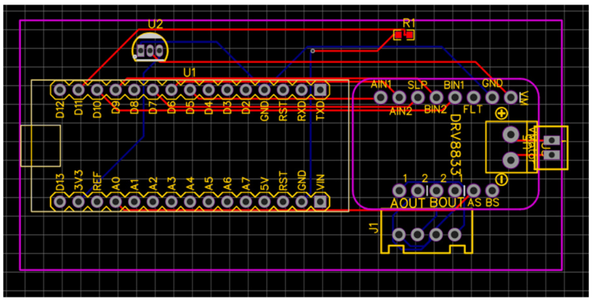

Figure 4: PCB schematics

IV. METHODOLOGY

The above system model was implemented, and the main manual operation was tested by the authors. The manual inputs come from a joystick ROS2 node. The joystick axis values are received by the ROS2 boat driver, and the data undergoes scaling operations to ensure compatibility with motors. Following the scaling procedure, data is encoded into a message sent to Arduino via UDP over Wi-Fi. Subsequently, the Arduino processes these received and scaled values to generate motor control instructions, thereby determining both the direction and speed of rotation required for the motors.

In parallel, the Arduino program reads the values from the gyroscopes and accelerometers, calculates its own attitude based on the values, and broadcasts them to the ROS2 driver node for further processing. The fusion of accelerometer and gyroscopes is performed by the Kalman filter running directly on Arduino. While roll and pitch angles can be estimated with bounded estimation error, the lack of magnetometers does not allow bounded estimation of the yaw angle. To calculate the current boat position, the accelerometer reading needs to be transformed into body velocity, which is in turn converted into pool frame velocities and integrated into pool frame position. Three problems are encountered: a) influence of gravity acceleration in sensor measurements, b) unknown yaw angle in pool frame, c) accrued error due to double integration of the attitude.

The first problem needs subtraction of gravity influence. To subtract gravity, based on the current roll and pitch estimates, the gravity vector is transformed into the IMU frame and subtracted from readings. The current boat roll and pitch are calculated, creating a gravity vector that is then subtracted from accelerometer readings. Following the calculation of pure linear boat acceleration, the linear acceleration itself is put through a dead zone filter to be more robust against Arduino's accelerometer inaccuracies as well as the Kalman attitude prediction error. The second problem and third problem are harder and require external aiding measurements from the camera. Since cameras are already calibrated in the pool frame, the measured heading of the boat marker corresponds to the boat heading in the pool frame and can be integrated into the Kalman filter as a correction. Similarly, the marker position from the camera can be used to bound the error accumulation due to double integration of the accelerometers.

For camera processing two ROS2 nodes were created. The first node deals with tracking of the boats. Tracking is based on ARUCO marker ([6]) tracking with 4x4-50 standard which codes 4x4 grid into 50 combinations allowing to track up to 50 different subjects. One example of the ARUCO code is shown in Fig. 3. The code is 3D printed and adjusted to stay within the width of the boat and be fixed on the removable boat cap.

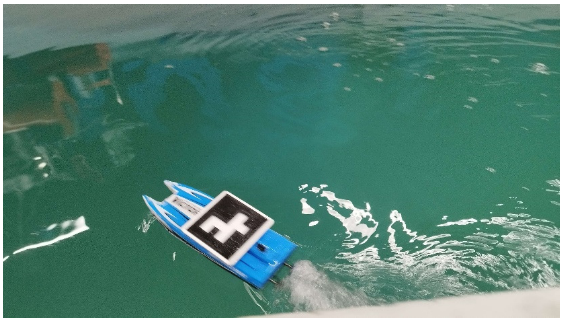

Figure 5: Example of the ARUCO marker mounted on the boat.

The initial plan for tracking was to use a coloured LED as a marker and a white LED to determine orientation. Multiple boats would be differentiated by the specific colour of the coloured LED. However, the problem with LED tracking was the weak detection of colours on LED diodes. The cameras would detect them as a white glow with a coloured afterglow ring, which was not consistently detectable. Additionally, there were also problems with the reflection of light from other objects. Hence, the choice of the ARUCO markers, which provide more consistent detection and identification of individual boats. Detection from ARUCO markers is robust until 30° of roll or pitch, for better camera resolutions or bigger marker sizes it works robustly until 40°. At 40° to 50° tracking moving object also becomes more difficult even on sufficiently good setups. In moderately wavey environment suitable for small vessel operated within a pool, steep angles are not expected. In cases where such angles occur the IMU should be able to sustain accurate tracking during temporary loss of camera tracking. Future testing will consider camera tracking performance with different wave frequencies and amplitudes to determine exact limits for our use case.

The second node for tracking is a poolside LED. The purpose of this LED is to estimate the camera acquisition delay, which can be attached to position and heading measurements for fusion in the Kalman filter running onboard the vessel. Delay detection is based on detecting the LED turning on (rising edge). The LED is operated by another poolside Arduino, which is in turn triggered by a signal from the camera processing computer. The ROS2 node triggers the Arduino LED via serial communication and measures the time it detects the LED turning on in the image. This detection is based on the OpenCV library and uses blob detection; the calculated delay is sent to the first ROS2 node to correct the measurement timestamp. The motivation for estimating the delay is that the used cameras are IP-based cameras connected to the busy lab network. This introduces a delay from 100-300ms. Additionally, the processing and ROS2 camera acquisition currently introduce an additional 300ms. Therefore, to fuse this in the Kalman filter, it is necessary to account for the delay since otherwise, the final state estimate becomes oscillatory. The delay is integrated into the Kalman filter by use recalculation. Once a delayed measurement arrives the sate is restored to the historical timestep when the measurement was valid, and the filter correction is performed. Following the correction the Kalman filter prediction is repeated until the current time step. The drawback of this approach is resource intensive as several Kalman filter cycles need to be performed within a single timestep. Alternate approaches, considering some assumptions about the model, can be utilized to reduce this workload However, these are left for future work by the authors at this stage.

V. RESULTS

Current results include operation on a single boat where the aim is to detect hardware, software, and general usability issues. The final component list on a single unit is shown in Table I.

TABLE I. BILL OF MATERIALS FOR THE TEST USV

Component	***Count***	***Price***
RC boat with thrusters (Conrad)	1	**€ 15,00**
Adafruit DRV8333 board (Mouser)	1	**€ 4,46**
Arduino Nano IoT 33	1	**€ 23,99**
MCP1700-3302E/TO	1	**€ 0,47**
Samsung 18650-S3000Q	1	**€ 5,19**

To connect the customized electronics, the first step was to remove the existing control electronics from the RC boat and solder the wires to the Arduino and DRV8333 boards. However, for better usability adding small connectors to the thruster outlets is better. Additionally, for the first prototype the Arduino and DRV8333 board were connected directly with wires but joining them on a single PCB with the basic connection traces and connectors for thrusters makes it much easier to exchange electronics when needed.

The Kalman filter for attitude estimation on the Arduino was tested for valid estimation, but delay compensation in the Kalman filter is still missing and therefore the position estimates are not stable.

Camera acquisition delay is correctly measured. The node can be configured with following parameters: a) sensitivity, b) minimum area, and c) region of interest (ROI) coordinates. The sensitivity and minimum area control the minimum and maximum blob size considered for rising edge detection. The ROI parameters is predefined since for each of the three cameras above the pool the LED is placed in the predetermined location. By selecting the ROI, the delay introduced because of camera processing is not considered and only the acquisition delay is measured.

The ARUCO based tracking operates as expected and the marker is robustly detected even with water surface reflections. Example of marker ID and orientation detection is shown in Fig. 4.

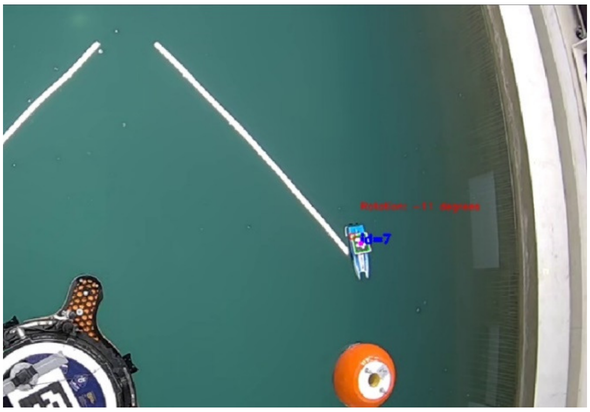

Figure 6: ARUCO marker recognition

VI. DISCUSSION

During the testing, one of the encountered issues was camera delay. To better integrate camera measurements with the onboard boat Kalman filter it is necessary to minimize the delay. In the ideal scenario the delay would be zero so delay minimization is an important next step as accuracy of the measurements would greatly improve with lower delay. The next step is to incorporate back calculation in the Kalman filter to allow proper integration of the delayed camera aided measurements. This will open the potential to implement control algorithms directly on the Arduino side and provide greater initial autonomy (e.g., be able to accept velocity commands from user).

From the perspective of inertial-only navigation an interesting research avenue would be to evaluate the minimum rate of external camera position aiding. Classical methods of calibration for the IMU could be performed to provide more accuracy. Additionally, by utilizing camera aiding as ground truth a data-driven model of the inertial sensors can be created that could capture non-systematic errors and prolong overall inertial-only operation towards a minute. However, the additional challenge is implementing these data-driven models on highly embedded targets such as the Arduino Nano IoT.

The current camera acquisition frame rate is 20 FPS, but an improved refresh rate of the camera would provide higher precision in position estimation. Namely, on lower frame rates, there is noticeable stretching on moving objects making it harder to gain accurate measurements from the marker. The tracking with ARUCO markers works on relatively slow objects and at the maximum speed, the boat could approach the limit for robust tracking. There exist tracking alternatives such as AprilTags that could potentially provide better performance. In the future, a comparison of marker methods will be investigated to select the best-suited for our use case. Additionally, to ensure robustness even for lower FPS, machine learning techniques for tracking will be investigated.

IMU hardware wise, the Arduino Nano IoT 33, which we are currently using as of time of writing, is quite convenient due to its size which alongside the driver perfectly fits inside the upper side of the deck on boat. Arduino Nano IoT 33 also has Wi-Fi module alongside accelerometer and gyroscope making communication with ROS2 central unit easy. While current model is good, it is missing a magnetometer which its “cousin” Arduino Nano 33 BLE contains, but which lacks built in Wi-Fi module. The gyroscope would be used in combination with current angel measurements to create an even better estimation of boat rotation, which would make entire state estimation more accurate if calibrated and calculated correctly. Other options that don’t include Arduino would be ESP8266 (for Wi-Fi capable processing unit) in combination with sensors such as MPU-6050 which contains accelerometer and gyroscope combo built into in. Magnetometers that have been considered are HMC5883L and QMC5883L which also fall into the similar price range of other components.

ACKNOWLEDGMENT

This work was supported by the project MASK. The project MASK - Marine Robots for better Sea Knowledge awareness has received funding from the EU Erasmus+ program, Key Action 2, 2022-1-HR01-KA210-SCH-000082831. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

REFERENCES

[1] Bell, Stephanie. “Project-Based Learning for the 21st Century: Skills for the Future.” *The Clearing House* 83, no. 2 (2010): 39–43. http://www.jstor.org/stable/20697896.

[2] N. Mišković, Z. Triska, Đ. Nađ and Z. Vukić, "Guidance of a small-scale overactuated marine platform — Experimental results," 2011 Proceedings of the 34th International Convention MIPRO, Opatija, Croatia, 2011, pp. 684-689.

[3] Ferreira, Fausto, Juraj Obradović, Đula Nađ, Ivan Lončar, Luka Mandić, Igor Kvasić, Natko Kraševac, and Nikola Mišković. 2024. "Remote-Access Marine Robotics Infrastructure and Experiments at LABUST" Journal of Marine Science and Engineering 12, no. 2: 317.

[4] Calvin Gregory, Andrew Vardy, MicroUSV: A low-cost platform for indoor marine swarm robotics research, HardwareX, Volume 7, 2020, https://doi.org/10.1016/j.ohx.2020.e00105.

[5] Wonse Jo, Yuta Hoashi, Lizbeth Leonor Paredes Aguilar, Mauricio Postigo-Malaga, José M. Garcia-Bravo, Byung-Cheol Min, A low-cost and small USV platform for water quality monitoring, HardwareX, Volume 6, 2019,

https://doi.org/10.1016/j.ohx.2019.e00076

[6] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. 2014. "Automatic generation and detection of highly reliable fiducial markers under occlusion". Pattern Recogn. 47, 6 (June 2014), 2280-2292. DOI=10.1016/j.patcog.2014.01.0