RoboBoat 2024: Technical Design Report SimLE SeaSentinel Team

Igor Rusiecki, Jakub Wilk, Cezary Wieczorkowski, Tomasz Ujazdowski, Serhii Pyskovayskyi
Marcin Solarczyk, Patryk Sobolewski, Karol Rzepiński, Paulina Prudel, Piotr Stroinski, Maciej Zawadzki, Wiktor Sieklicki
Gdańsk University of Technology, Poland, Gdańsk

***Abstract*—This report discusses the strategy of a SimLE SeaSentinel Team for RoboBoat 2024 and the design of our system. The system consists of an Autonomous Surface Vehicle (ASV) named ASV Rybitwa, a Ground Segment and a set of procedures that allow us to effectively operate it all. In our work, we try to follow systems engineering principles. For this reason our work during the last ten months consisted of following phases: mission definition and general strategy, requirements, concept, design, production and testing. We have come to a conclusion that the best course of action for our team would be to try to approach the same tasks as last year. In essence, we decided that ASV Rybitwa will be an improved version of ASV Perkoz as most of or main design features proved to work very effectively during RoboBoat 2023 competition. ASV Rybitwa will be a modular, easily transportable catamaran made mostly of fibreglass and 3D printed PET-G. The propulsion system consisting of four thrusters will provide us with omnidirectional movement capabilities. In addition, we will equip our ASV with a water cannon in order to approach Task 4. All modules within our boat will communicate with each other via an Ethernet network. With the help of our new partner AQ Wiring Systems STG, we also managed to develop a more professional electrical system. Computer vision will rely solely on OAK-D stereo cameras. As a basis of our software system architecture, we decided to utilize Robot Operating System 2 (ROS2). We will maintain communication with our ASV with the help of three different radio links. Apart from the boat, we also decided to improve our Ground Segment, which includes a brand new Operator Control Station and modified transport and handling equipment. In addition, we greatly increased the amount and frequency of conducted tests.**

***Index Terms*—autonomous surface vehicle, robot operating system, behavioural trees, omnidirectional propulsion, RoboBoat**

Acronyms and Abbreviations

ASV Autonomous Surface Vehicle
VCU Vehicle Control Unit, a.k.a. Flight Control Unit
HFOV Horizontal Field of View
COCO Common Objects in Context
VPU Vision Processing Unit
OCS Operator Control Station
OBP Onboard Processing
DOF Degrees of Freedom
SBC Single Board Computer
ROS2 Robot Operating System 2
DMS Decision-Making System
GS Ground Segment
FOV Field of View
UDP User datagram protocol

I. Competition Goals

A. General Strategy

In our work, we try to follow rules established in the field of systems engineering. One of them is the V-model of Systems Engineering Process (Fig.1)

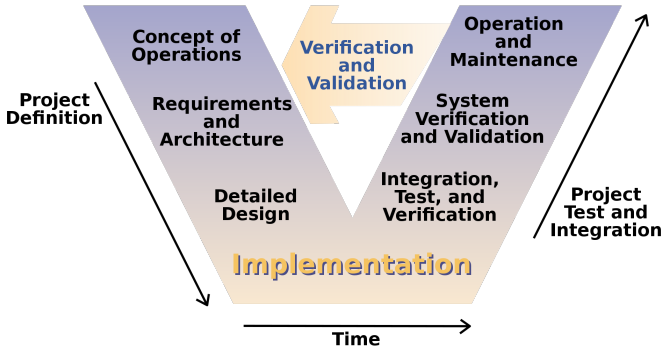


Fig. 1. The V-model of the systems engineering process. [1]

The process of developing ASV Rybitwa can be divided into the following phases:

- **Lessons Learned** - Right after returning from the USA we organized a couple of meetings during which all team members shared their thoughts, feelings and general feedback regarding the way we conducted our mission and the functioning of ASV Perkoz (our previous vehicle). That is how we created an organized list of lessons that we have learned during our time in Florida. This document later became a basis for an After Action Report.
- **After Action Report** - In May 2023 we summarized the entire process of building ASV Perkoz, our accomplishments during RoboBoat 2023, all the important lessons learned and plans for the future in a single presentation. We delivered our presentation to our stakeholders from both university and industry.
- **Mission Analysis** – During our Mission Analysis, we have come to a conclusion that the most optimal course of action for our team will be to approach the same tasks that we set out to approach during RoboBoat 2023. Last year, despite our efforts, we managed to approach only two

tasks during finals. It did fit into our definition of success, especially since it has been our first time participating in RoboBoat. Yet, it was the absolute minimum of what we wanted to achieve during the previous edition of this competition.
For this reason, we concluded that expanding the scope of our project beyond the six tasks that we tried to approach last year would stretch our efforts too thin. Another limiting factor was the time available for development, or rather the lack of it. In 2024 the competition takes place over a month earlier in comparison to 2023 which put additional pressure on the team to deliver the system as soon as possible.

- **Requirements** – Law no. 13 from Akin's Laws of Spacecraft Design states that: "Design is based on requirements. There's no justification for designing something one bit "better" than the requirements dictate." [2] Creating requirements is an absolutely critical element of the entire process, and thus we decided to pay a lot of attention to it and spent considerable amount of time to properly define them. Many cost overruns and delays are caused by over-ambitious or missing requirements [3]. In order to set our priorities and formulate requirements for ASV Rybitwa we decided to use the MoSCoW method [4]. We identified requirements related to Tasks 1, 2, 3 and 8 as "must-have", requirements related to Tasks 5 as "should have", Task 4 related requirements as "could have" and finally we identified Task 6 and 7 related requirements as "won't have". Altogether, we came up with a requirement tree that contained over one hundred single requirements ranging from the most general (i.e. it must move on the water surface) to very specific ones (i.e. buoyancy must not be provided by confined spaces filled with air). We passed Preliminary Requirements Review in June and Critical Requirements Review in late August.
- **Concept** – In September, we began working on the general concept of our system. Our goal was to answer the most basic and fundamental questions like: "will it be a catamaran or a mono-hull?", "what equipment will computer vision be based on?", "what will be the source of propulsion?" etc. We entered a partnership with KNW "Proces" from Beautiful Arts Academy in Gdańsk. Our desire was to improve the aesthetics of our design and create concept arts (Fig. 2) that would be an inspiration for engineers during various phases of development. We ended this phase of development with a Preliminary Design Review.
- **Design** – After agreeing on the general idea of what our system should look like and how it should function, we began the design phase. Details regarding this phase are described is Design Strategy section. Due to limited time frame and delays in some areas, we decided to begin production of certain subsystems before Critical Design Review of the entire system. This deviation from the V-model comes with its own risks, but was necessary to accelerate the process.

Fig. 2. Concept art by Aleksander Kwiek from KNW "Proces"

- **Production** - Last year we limited ourselves mostly to 3D printing since it was the only manufacturing technology that we were well accustomed with. This year, however, with the help of KSTO Korab we began to introduce fibreglass to our project. We still utilize 3D printing to a great extent as a cost-effective method of manufacturing non-standard components. Thanks to our partnership with AQ Wiring Systems STG, we also moved away from the idea of 3D printing boxes. Instead, we began to use industry standard, waterproof boxes provided by our partner. AQ also provided us with prefabricated components that greatly improved both simplicity and reliability of our electrical system.
- **Testing** - According to the V-model, validation and verification takes place during every step of the design process. For this reason, "Testing" cannot be considered to be a separate, subsequent phase. It was happening in various ways (test fits, models, simulations, live tests) in parallel. We have found this approach a major improvement in comparison to our previous works on ASV Perkoz.

B. Course Strategy

The only significant change to our Course Strategy in relation to RoboBoat 2023 is that we abandoned the idea of using a large robotic arm to deliver a steady stream of water to the target. This solution proved to be overly complicated especially when it comes to software.

Firstly, the ASV will attempt the mandatory **Task 1 - Navigation Channel**. For every task, the ASV's motion control will be based on computer vision and the decision-making process will be supported by a behaviour tree.

After completing the first task, the ASV will detect the pair of green and red buoys, which will indicate the end of Task 1 and the beginning of **Task 2 – Follow the Path**. Object avoidance will be based on image acquisition and processing, and the ASV will be suggested to move to the nearest "clear" segment. Furthermore, distances to objects will be calculated based on the data provided by three OAK-D stereo cameras. Total Field of View (FOV) is 207 degrees. To minimize the risk of the ASV getting lost, we will save its GPS coordinates

before and after every task. In case of being unable to detect any desired object, ASV will return to the last saved position.

For **Task 3 – Docking** – we are training our model to detect posters with different shapes (circle, triangle, duck) in three colours (green, blue and yellow). Basing on the given colour and shape, the ASV will detect the object. To dock, we will use path planning and object avoidance algorithms.

The core of our strategy for **Task 4 – Duck Wash** – is the use of our custom Water Cannon capable of movement in two DOFs and delivering a steady stream of water to a designated target. After detecting a poster with a picture of a Duck, our ASV will move to this bay and position itself in close proximity to the target, but not too close in order to not lose sight of the Duck. After that, the water cannon will begin to spray the Duck with water. In order to successfully execute Task 4 this way, we need to ensure that we will be able to maintain a stable position and minimize any leeway. For this reason, we equipped our ASV with a four-thrusters propulsion system to enhance our capabilities regarding dynamic positioning.

While preparing our strategy for **Task 5 – Speed Challenge** – we analysed both videos from previous editions of RoboBoat and our own experiences from RoboBoat 2023. We have come to a conclusion that the main factor determining the success is not speed but the ability to fluently turn around the blue buoy. As an old saying says: "Slow is smooth, smooth is fast". Speed will not matter if our boat gets lost. This is why we did not identify the speed of our vehicle to be a critical parameter, although our thrusters can generate up to 44 N of thrust each [7]. Instead, we put grater emphasis on wide FOV and mapping of detected objects.

Task 8 - Return to Home is the last task on the entire course and also the last that we intend to approach. Our plan for this task is to use the proximity of Task 8 black buoys to Task 1. Before entering the Navigation Channel, our ASV will take a look around itself in order to detect, identify and map the location of black buoys. Black buoys from Task 2 will not be mistakenly identified as buoys from Task 8 as we will include a condition in our algorithms that Task 8 buoys cannot be further than approx. 12 m from ASVs starting position. Later on it will use this information to move to the area in which it detected black buoys from Task 8 while avoiding all potential obstacles. When Task 8 buoys will be within visual range of our boat it will start to rely on its cameras to once again detect, identify and locate these buoys and reach the target.

II. Design Strategy

As stated before, since the very beginning of SeaSentinel's existence, we have adhered to the principles of systems engineering. We divided our project into subsystems and components, which have been assigned to team members for development and implementation. Each top-level system has been defined with a specific role and function, according to single responsibility principle. We have chosen to split our work into the following systems:

- Mechanical – responsible for hull and superstructure design.
- Special Task Modules - holistic development of task specific modules, i.e. water cannon.
- Vehicle Control Unit – Pixhawk and PX4 configuration, thruster configuration and GPS setup.
- Electrical – all the cabling, batteries, power, and integration of Vehicle Control Unit (VCU) (a. k. a. Flight Controller) with sensors and motors.
- Onboard Processing (OBP) – application domain, high-level command over VCU, object detection, decision-making, task-specific algorithms, simulation.
- Ground Segment – everything that won't be on the Autonomous Surface Vehicle (ASV), including the Operator Control Station (OCS).
- Telecom - radio links, communications between ASV and OCS.

Their structure and components have been reflected within our Work Breakdown Structure (C-A). Detailed descriptions of some of the components have not been mentioned in this report in order emphasize the creative aspects of our design.

A. Hull design

Our team decided that ASV Rybitwa, just like ASV Perkoz will be a catamaran. Another option that we took into considerations was a flat bottom barge-like mono-hull shape. A flat bottom boat would me more difficult to tip over and thus safer. However, flat bottom boats are more prone experience significant rolling. Our last time in Florida convinced us that strong winds and waves up to 20 cm high are something that we need to take into account. The more intense the rolling of the boat, the more difficult it will be for our cameras to correctly identify their surroundings. That is why we decided to build a catamaran which still will be very stable and will have the ability to pierce through waves instead of swaying on them.

Last year in Florida we experienced some difficulties with our hull design. After reaching certain speed our bow had a tendency to submerge when moving forward. Same would happen for the aft when moving backwards. We conducted tests on Gdańsk University of Technology Testing Pool to establish what could be the reason behind it. We concluded that at approx. 1 kt the wave created by the movement our hull is so high that it floods the fore deck (Fig. 3). This additional weight of water on the deck resulted in an undesirable trim in the direction of movement.

The above led us to a conclusion that we need to develop a new hull shape that will generate less hydrodynamic drag and will break waves to the sides instead of letting them flood the fore deck. With the help of Cezary Żordowski, PhD we understood that designing an entirely new hull shape would unnecessary considering the size of our ASV. Instead we based our design on the design given to us by Mr. Żrodowski. We cut it in amidships and mirrored in order to make bow and aft symmetrical. As a result the hull is equally efficient when moving forward and backwards which corresponds with our

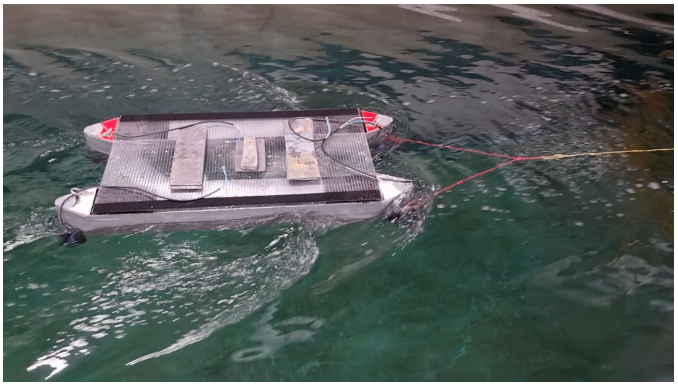


Fig. 3. Hull drag tests

requirements regarding omni-directional movement capabilities. We changed the ratio of main dimensions (length, breadth, height) to fit our requirements regarding displacement and maximal length overall. Another feature implemented by us in hull design are slots for columns to which thrusters will by mounted (Fig. 4).

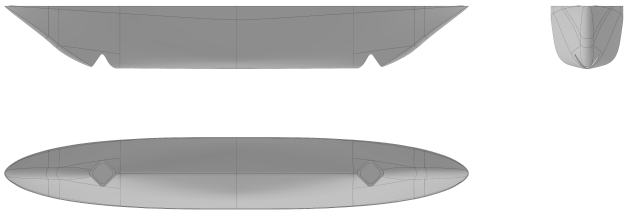


Fig. 4. Hull design with cutout slots for thruster columns

The hull has been made out of fibreglass with the use of vacuum infusion technology. It has been filled with a closed cell, expanding polyurethane foam. Last year's sinking of another team's boat led us to believe that buoyancy of our ASV should not be provided by an enclosed space filled with nothing but air but rather by an enclosed space filled with material capable of displacing water even in an event of hull puncture.

B. Propulsion

Similarly to our previous vessel, ASV Rybitwa will be equipped with a differential thrust propulsion system with the use of four Blue Robotics T200 Thrusters [7]. After testing, this setup during RoboBoat 2023 we determined that it offers great agility and excellent dynamic positioning capabilities.

To control this propulsion setup, we are using our custom controller that extends PX4 autopilot firmware [6]. We were improving this controller for the past months to increase its stability and utilize the full potential of our thruster configuration.

C. Water Cannon

Water Cannon is the centerpiece of our strategy to approach Task 4. Opposite to last year, during this project we focused on simplicity. The cannon can move in two DOFs (elevation and horizontal rotation). It consists of two servo motors, little plastic tube, a set of bearings and a 3D printed structure to hold it all. There is a separate module for water pumps in order to prevent any potential damage to electrical components due to pump leakage.

For driving, we use inverse kinematics. This method is coupled with the camera feedback. As a fallback, we have hard-coded the desired arm position and lock it while executing the task, disregarding camera feedback.

D. System Bus

For ASV Rybitwa, we decided to take a different approach to communication between various systems on our boat in comparison to our previous design. On ASV Perkoz we utilized CAN bus and OpenCyphal [22] protocol which proved to be hard to integrate with Robot Operating System 2 (ROS2) [14]. This time we desired a more unified system. We decided to use an Ethernet network to connect all of our subsystems as shown in Appendix C-D. Ethernet has some disadvantages, mainly the fact that it is a centralized network, which means that there is a single point of failure. But the main advantage of Ethernet is its wide adoption. Our VCU and Single Board Computer (SBC) running all autonomy software already have Ethernet interfaces and we easily found an Ethernet enabled microcontroller board to use in our hardware modules [21].

Easy integration of our hardware with ROS2 was our main criteria when choosing a communication protocol. For this reason we picked a protocol from creators of ROS called micro-ROS [20]. It aims to help with interfacing ROS 2 applications with hardware and enables microcontrollers with limited resources to publish and subscribe to topics on a ROS network. Furthermore, it supports Ethernet as a physical transport protocol. It enables us to seamlessly integrate our hardware modules into our autonomy stack.

E. Electrical system

This year, we partnered with AQ Wiring STG to bring our electrical system to a more sophisticated level. Similarly to ASV Perkoz, ASV Rybitwa uses Li-Po batteries as a source of power. We kept a dedicated power management module to handle power sequencing, monitoring and remote shutdown via separate radio link. After visiting AQ's headquarters, we took inspiration from various automotive electrical systems design they manufacture and decided to structure ours similarly. We put every module on a separate fused circuit like shown in Appendix C-B. This might be considered a bit excessive for such a small vessel but it makes the whole system safer and also makes troubleshooting possible faults much easier.

One of our biggest goals was to mitigate high electromagnetic interference. Main sources of this interference were thruster controllers and cameras' cables. This interference caused us many problems with the GPS signal during last

year's competition. With the help of experts from AQ, we were able to choose suitable shielded cabling and enclosures that minimized interference onboard our vessel.

F. Position and attitude determination

GPS RTK (Real-Time Kinematic) is a high-precision positioning technology that uses differential correction to improve the accuracy of GPS measurements by several orders of magnitude, allowing for centimetre-level accuracy in real-time. Additional data is being fed passively to ASV from OCS during its autonomous operation, as it has an integrated GPS RTK base station.

To increase accuracy of heading determination, we will attempt to use a second GPS module to supplement VCU's compass readings. Each module will be positioned on each board, symmetrically in relation to Center Line.

G. Computer Vision

For computer vision, we decided to use OAK-D [13] stereo cameras, as we found them affordable and sufficient while working on the previous iteration of our ASV. These cameras feature a Vision Processing Unit (VPU), as well as colour and stereo vision capabilities. The colour data is fed to the neural network model running on the SBC, while the stereo data is processed on the cameras themselves. This allows the SBC to be unburdened of additional workload and have more resources available for autonomy. To achieve wider spatial awareness, we chose to use three cameras totalling 207 degrees of Horizontal Field of View (HFOV). Both object detection and distance data are transmitted as ROS2 topics.

The neural network architecture we selected for performing object detection task is YOLOv8 [11], which is the latest edition of the YOLO family [12]. In comparison to YOLOv7 that we used previously, it proved to work faster and return more accurate detections. In order to reduce the usage of computation resources, we opted to utilize a nano version of YOLOv8 with fewer trainable parameters, pre-trained on a Common Objects in Context (COCO) dataset. This is required due to constrained resources on our chosen SBC: Jetson Nano. It has been trained using data that we gathered during Roboboat 2023 and labelled using labelImg [8] which supports the YOLO architecture.

H. Autonomous Navigation Software Architecture

Suppose $\mathbb{R}^n$ is to represent a vector space over a field of real numbers $\mathbb{R}$ with usual addition $(+)$, scalar multiplication $(\cdot)$, Cartesian product $(\times)$, and $\mathbb{T} \subset \mathbb{R}$ denotes an open set. Taking a set of elements from a positive part of an integer field, $\{n_\mathrm{x}, n_\mathrm{u}, n_\mathrm{d}, n_\mathrm{y}, n_\mathrm{c}\} \subset \mathbb{Z}_+$, enables one to construct a n-tuple of vector valued functions $\mathbb{T} \to \mathbb{R}^n$ such that $\forall t \in \mathbb{T}$ it follows that $(\boldsymbol{x}(t), \overline{\boldsymbol{u}}(t), \boldsymbol{d}(t), \boldsymbol{y}(t), \boldsymbol{c}(t)) \in (\mathbb{X}_\mathrm{x}, \mathbb{X}_{\overline{\mathrm{u}}}, \mathbb{X}_\mathrm{d}, \mathbb{X}_\mathrm{y}, \mathbb{X}_\mathrm{c}) \subset (\mathbb{R}^{n_\mathrm{x}}, \mathbb{R}^{n_\mathrm{u}}, \mathbb{R}^{n_\mathrm{d}}, \mathbb{R}^{n_\mathrm{y}}, \mathbb{R}^{n_\mathrm{c}})$. In particular, t is to denote time.

$$\Sigma_\mathrm{ASV} : \begin{cases} \mathbb{X}_\mathrm{x} \times \mathbb{X}_{\overline{\mathrm{u}}} \times \mathbb{X}_\mathrm{d} \to \mathbb{R}^{n_\mathrm{x}} \\ \mathbb{X}_\mathrm{x} \times \mathbb{X}_{\overline{\mathrm{u}}} \times \mathbb{X}_\mathrm{d} \to \mathbb{X}_\mathrm{y} \\ \mathbb{X}_\mathrm{x} \times \mathbb{X}_{\overline{\mathrm{u}}} \times \mathbb{X}_\mathrm{d} \to \mathbb{X}_\mathrm{c} \end{cases}, \tag{1}$$

where $\mathbb{X}_\mathrm{x}$, $\mathbb{X}_{\overline{\mathrm{u}}}$, $\mathbb{X}_\mathrm{d}$, $\mathbb{X}_\mathrm{y}$, $\mathbb{X}_\mathrm{c}$ are used to denote operating regions in the state, control and disturbance input, measured and controlled output spaces, respectively.

The measurement system providing GPS, IMU, and Computer Vision (Section II-G) measurements is given by:

$$\Sigma_\mathrm{M} : \mathbb{X}_\mathrm{y} \times \mathbb{X}_\mathrm{d} \to \mathbb{X}_{\mathrm{y_m}}. \tag{2}$$

The control signals affect the ASV through an actuators system, consisting of a multi-propeller system (Section II-B) as:

$$\Sigma_\mathrm{A} : \mathbb{X}_\mathrm{u} \to \mathbb{X}_{\overline{\mathrm{u}}}, \tag{3}$$

where $\mathbb{X}_{\overline{\mathrm{u}}} \subseteq \mathbb{X}_\mathrm{u} \ni \boldsymbol{u}(t)$, $\forall t$.

The multi-propeller system is under the control of the Vehicle Control Unit Σ_VCU, which, $\forall t$, maps measurements $(\boldsymbol{y}_\mathrm{m}(t) \in \mathbb{X}_{\mathrm{y_m}} \subset \mathbb{X}_\mathrm{y})$ and reference trajectories $(\mathbb{X}_\mathrm{r})$ into control signals:

$$\mathbb{X}_\mathrm{u} \leftarrow \Sigma_\mathrm{A} \circ \Sigma_\mathrm{VCU} \circ \Sigma_\mathrm{M} \left[\mathbb{X}_{\mathrm{y_m}} \times \mathbb{X}_\mathrm{r}\right], \tag{4}$$

where $\circ$ denotes a composition operator.

The role of the Decision-Making System (DMS) is to assign mission and submission objectives $(\boldsymbol{r}_\mathrm{MO})$ to ASV under supervision based on mission status reports $(\boldsymbol{r}_\mathrm{MR} \in \mathbb{X}_\mathrm{MR} \subset \mathbb{X}_\mathrm{y})$. The reference trajectory to be realized by the Σ_VCU is generated by a sophisticated system Σ_DMU, based on measurements and mission objectives, and described as:

$$\Sigma_\mathrm{DMS} : \boldsymbol{r}_\mathrm{MO} \times \mathbb{X}_\mathrm{MR} \mapsto \mathbb{X}_\mathrm{r},$$

where: $\mathbb{X}_\mathrm{r}$ is a set of admissible mission objectives, that enables considered surface vehicle unit (Σ_ASV) to carry out an autonomous execution of prescribed mission objectives.

Therefore, it can be indicated that the control system is divided into a higher control layer for strategy development and a lower control layer for trajectory execution and communication with peripherals.

Within the higher control layer (Σ_DMS) two algorithms were used. The first is Behavioural Trees (BT) [10] used to make decisions based on mission status and mission objectives. The second algorithm is Artificial Potential Field (APF) used to determine optimal paths to the current navigation destination taking obstacle avoidance into account. The lower control layer (Σ_VCU), uses PX4 [6] autopilot firmware.

The Σ_DMS has been implemented using the ROS2 using packages such as Nav2. This choice of framework allows seamless integration with a variety of sensors and actuators, enabling efficient data acquisition and digital filtering. The node-based architecture adopted in ROS2 promotes code clarity, facilitates testing and increases the flexibility and extensibility of the system.

In addition, communication between Σ_DMS and Σ_VCU has been established using the uXRCE-DDS [15]. This middleware allows effective data exchange with the PX4 autopilot via ROS 2 topics. This approach ensures a consistent data structure across all control layers, contributing to the overall modularity and scalability of the system.

This design choice is in line with the project's objective to create a modular and scalable ASV capable of resilient operation in the event of individual node failures. The use of ROS2 and uXRCE-DDS builds on the rich ROS package ecosystem, allowing core functionality to be developed while seamlessly integrating with existing libraries and tools.

I. Telemetry and remote shutdown

One of our biggest setbacks during the previous year's competition were problems with radio connectivity to our boat. For RoboBoat 2024 we decided to use a dedicated 2.4 GHz link for manual control, a 5 GHz link for autopilot telemetry and video feed and a 915 MHz link for emergency remote shutdown. Our manual control link utilizes ELRS [17] compatible RC controller and radio. ELRS is a protocol used for controlling RC planes or drones. As a result it is supported by our autopilot firmware [6] and offers excellent communication range. Our remote shutdown link will utilize simple 866MHz/915MHz radios that proved to be easier to source than reliable 433MHz modules. In order to avoid accidental shutdowns due to communication errors, we also decided to implement a simple protocol utilizing CRC and receive acknowledgments.

We decided to get rid of the 433MHz telemetry link, since it required additional hardware and proved to be very unstable. Instead, we will be using an IP telemetry link. Since the ASV is equipped with an onboard Ethernet network we can utilize Mavlink [23] over User datagram protocol (UDP) to send telemetry data across our Wi-Fi link. In order to use it, we must first ensure that our link is very reliable. For this reason we decided to use a dedicated Wi-Fi radio on the ASV's side. Last year we were using only a USB to Wi-Fi adapter, and it proved to be unsuitable for this task. Since both of our radios come from Mikrotik we decided to utilize their proprietary Nv2 protocol [19]. It provides a reduced propagation delay overhead and per frame overhead in comparison with standard 802.11 protocols. The above ensures that this mission critical radio connection is as reliable as possible.

J. Ground Segment

Our intention was to integrate our OCS hardware into one module. With this approach, we aimed to reduce setup times and simplify setup procedures. We came up with a design that fits all supporting OCS hardware (power supplies, router, etc.) into a protective case [16] with custom inserts that break out necessary connectors. The only external components are a laptop for running QGround control software and a Wi-Fi radio, both connected to our case with Ethernet.

III. Testing Strategy

Depending on the subsystem, we are able to test it either in a simulated environment or on a real unit. Due to long procurement time for parts and difficulties testing some components on an actual ASV, we developed a simulation using Gazebo [18] environment, rich PX4 [6] ecosystem and readily available docker containers [25]. This solution makes it faster and more comfortable to test and debug our software. It proved to be efficient during development of behaviour trees, navigation algorithms and communication between ROS2 and PX4. It also plays an essential role in the process of creating synthetic data for training our neural network. This type of data allows us to train YOLOv8 for detecting objects that were not previously present on the competition or would be hard to obtain. Moreover, it allows more team members to actively develop OBP in shorter period of time. Since the simulated environment does not contain elements that are less predictable, such as noise or interference, some subsystems cannot be tested in simulation. To solve this issue, we decided to use our previous ASV as a testing platform for such components. During water testing, we found out about many communication issues that would not be possible if we were testing our system only in the simulation. A major advantage of this solution is that besides testing technical aspects of the system, we also developed a more organized way to prepare our ASV for launch. It results in less time spent preparing for testing during the competition. We also purchased a set of buoys for testing neural network and algorithms on water (Fig. 5).

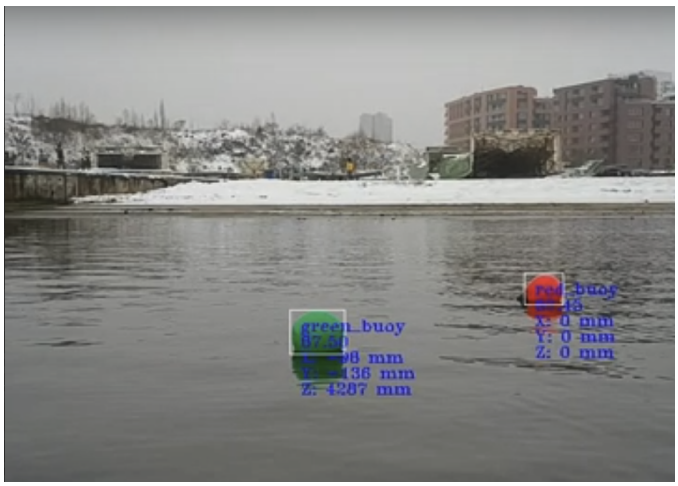


Fig. 5. Neural network detections during water testing

When it comes to testing individual hardware modules, we decided to stick to our FlatSat [24] inspired approach from the previous year. It is a testing method where each subsystem is laid out and connected to power and communication buses on a test bench. It allows us to test each subsystem in a controlled environment that simulates operational conditions. Another benefit of everything being laid out is that it is much easier to troubleshoot errors and make modifications to the setup in comparison to a situation when everything is tightly packed in the hull.

Acknowledgments

SimLE SeaSentinel Team would like to thank:
Wiktor Sieklicki, PhD for being our faculty advisor and mentor,

Norbert Szulc, M.Sc. for his insight, advice, support during tests and ongoing help with VCU's development and troubleshooting,
KNW "Proces" for creating our Mission Patch and Concept Arts of the vehicle,
CINESTAN for their help with video production,
Krzysztof Cwalina, PhD for his advice regarding wireless communication,
Jakub Zdroik, M.Sc. for his help and advise regarding technical aspects of our ASV,
AQ Wiring Systems STG for providing resources and expertise that helped design the electrical system of our vessel,
Hackerspace Pomerania community for sharing components, tools and their knowledge,
KSTO "Korab" and Piotr Pruszko for their help with hull manufacturing,
F.D.C. Willard for long and contributing discussions.

REFERENCES

[1] Osborne, Leon F., et al. *Clarus: Concept of operations.* No. FHWA-JPO-05-072. United States. Federal Highway Administration, 2005.
[2] Akin's Laws of Spacecraft Design, retrieved from https://spacecraft.ssl.umd.edu/akins_laws.html
[3] Olivier L. de Weck, Fundamentals of Systems Engineering: Requirements Definition, Lecture Notes, 2015, Available online: https://ocw.mit.edu/courses/16-842-fundamentals-of-systems-engineering-fall-2015/.
[4] Brennan, Kevin, ed. *A Guide to the Business Analysis Body of Knowledge.* Iiba, 2009.
[5] Ren, Lei, et al. *Environment influences on uncertainty of object detection for automated driving systems.* 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, 2019.
[6] Open Source Autopilot for Drones - PX4 Autopilot, retrieved from https://px4.io/.
[7] T200 Thruster: ROV thruster for marine robotics propulsion, retrieved from https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/.
[8] LabelImg - image annotation tool, retrieved from https://github.com/HumanSignal/labelImg. Access date 10 December 2023.
[9] Ujazdowski T. and Kułaga D. *Project and implementation control algorithm based on fuzzy logic for the yacht model*, 2021, Master's degree thesis, Gdańsk University of Technology, Poland
[10] BehaviorTree/BehaviorTree.CPP: Library in C++. Batteries included, retrieved from http://github.com/BehaviorTree/BehaviorTree.CPP
[11] Reis, Dillon, et al. *Real-Time Flying Object Detection with YOLOv8.* arXiv preprint arXiv:2305.09972 (2023).
[12] Redmon, Joseph, et al. *You only look once: Unified, real-time object detection.* Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[13] OAK-D — DepthAI Hardware Documentation 1.0.0 documentation, retrieved from http://docs.luxonis.com/projects/hardware/en/latest/pages/BW1098OAK.html
[14] Stanford Artificial Intelligence Laboratory et al. (2018). Robotic Operating System. Retrieved from https://www.ros.org
[15] eProsima Micro UXRCE-DDS - Overview, retrieved from https://micro-xrce-dds.docs.eprosima.com/en/latest/introduction.html
[16] Protective cases used https://www.auer-packaging.com/us/en/Protective-cases-Pro/CP-4316.html
[17] ExpreLRS protocol overview, retrieved from https://www.expresslrs.org/
[18] Gazebo showcase, retrieved from https://gazebosim.org/showcase
[19] Mikrotik's Nv2 protocol user manual, retrieved from https://wiki.mikrotik.com/wiki/Manual:Nv2
[20] Micro ROS overview, retrieved from https://micro.ros.org/docs/overview/features/
[21] Microcontroller board used https://www.wiznet.io/product-item/w5500-evb-pico/
[22] OpenCyphal documentation, retrieved from https://opencyphal.org/
[23] Mavlink protocol overview, retrieved from https://mavlink.io/en/
[24] Herbert J. Kramer. FlatSat (ground-based testbed for CubeSats). Available online: https://www.eoportal.org/other-space-activities/flatsat\#overview
[25] Docker overview. Available online: https://docs.docker.com/get-started/overview/
[26] Development containers. Available online: https://containers.dev/

APPENDIX A Component list

Component	Vendor	Model	Specs	Custom/Purchased	Cost	Year of purchase
ASV Hull	Custom	Catamaran	B=60cm L=130cm H=60cm	Custom	$1500	2023
Waterproof connectors	TE connectivity	Deutsch DT	te.com/usa-en/products/connectors/automotive-connectors/intersection/deutsch-dt-series-connectors.html	Purchased	$15	2023
Propulsion	Blue robotics	T200	bluerobotics.com/store/thrusters	Purchased	$236	2022
Power system	Redox	Li-Po battery	14.8V, 4400mAh	Purchased	$58	2022
Motor controls	Blue robotics	Basic ESC	https://github.com/bitdump/BLHeli	Purchased	$36	2022
Processing computer	Nvidia	Jetson Nano	developer.nvidia.com/embedded/jetson-modules	Purchased	$340	2023
Teleoperation	MikroTik	Groove AC	mikrotik.com/product/RBGrooveGA-52HPacn	Purchased	$120	2023
Cameras	Luxonis	OAK-D	store.opencv.ai/products/oak-d	Purchased	$249	2022
Vehicle control unit	Holybro	Pixhawk 5X	holybro.com/products/pixhawk-5x	Purchased	$300	2023
GPS	SparkFun	GPS-RTK2	sparkfun.com/products/15136	Purchased	$275	2023
Vision	-	Yolov8	yolov8.com	-	-	-
Autonomy	Open robotics	ROS	ros.org	-	-	-
Open source software	Open robotics	Micro ROS	micro.ros.org	-	-	-

APPENDIX B
TEST PLAN & RESULTS

A. Testing scope

We created test goals based on our experience from RoboBoat 2023. We wanted to water test our vision processing pipeline and motor control pipeline. We also wanted to water test our new wireless communication solutions.

B. Schedule

We have begun our water testing in November 2023 and plan to continue testing until our departure to the end of January 2024. We schedule our tests on weekends since that is when most team members are available. We have tests planned every week, but that is flexible since sometimes a week is not enough to make the necessary improvements so two days before each planned test our team leader makes sure that we have improved versions of software and hardware to test, if not the tests are rescheduled for another day.

C. Environment

Finding the suitable environment for water tests proved tricky for us. The first option was the tow tank basin at our university. Unfortunately, this testing environment is only viable for hydrodynamics or manual control testing since there is no GPS reception there. As a result we can not test autonomous operations there. Unfortunately, since our water tests begun during winter, all closed water bodies were frozen. For this reason, we moved on to test our ASV in seawater, specifically at Imperial Shipyard, which is an unused, open to public part of Gdańsk shipyard. An example of the testing environment is shown below.

D. Risk Management

Unfortunately, an old shipyard is not the prefect testing environment. Water is sometimes very shallow and contains various steel implements that could damage our vessel. This water body is also connected to the larger area of operational shipyard, which means that in case of manual control loss, our vessel could float away and be really hard to recover. To combat these issues, we utilized a few methods:

- ASV has to be tied to shore with sufficiently long rope to aid in recovery in case of sinking or losing manual control.
- One of our team members is always on the water in a raft that can tow our ASV back to shore.
- We utilize two or three spotters that communicate with the ASV operator via walkie-talkies and inform him of possible obstacles or dangers.

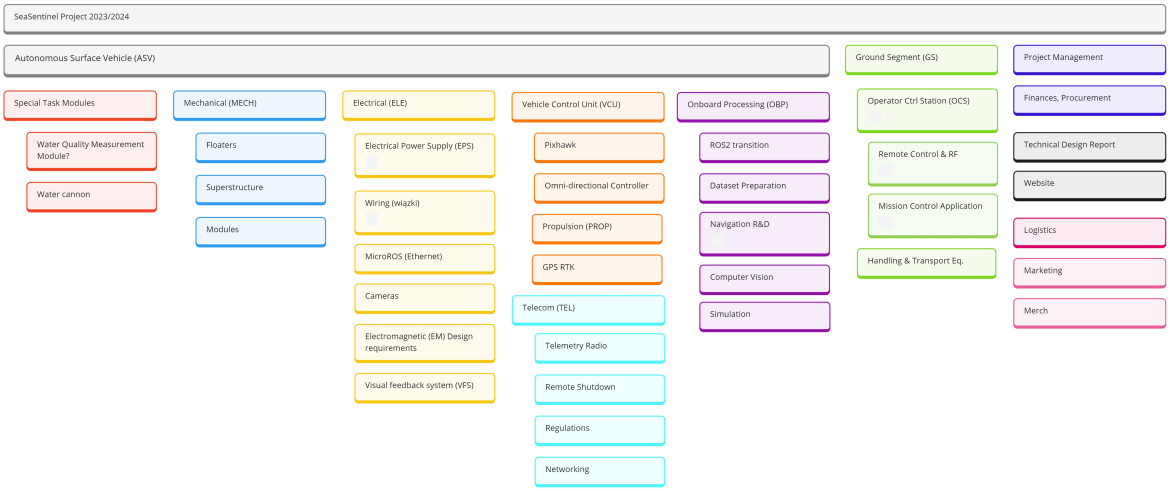
E. Results

Testing conducted up to this point resulted in the following conclusions:

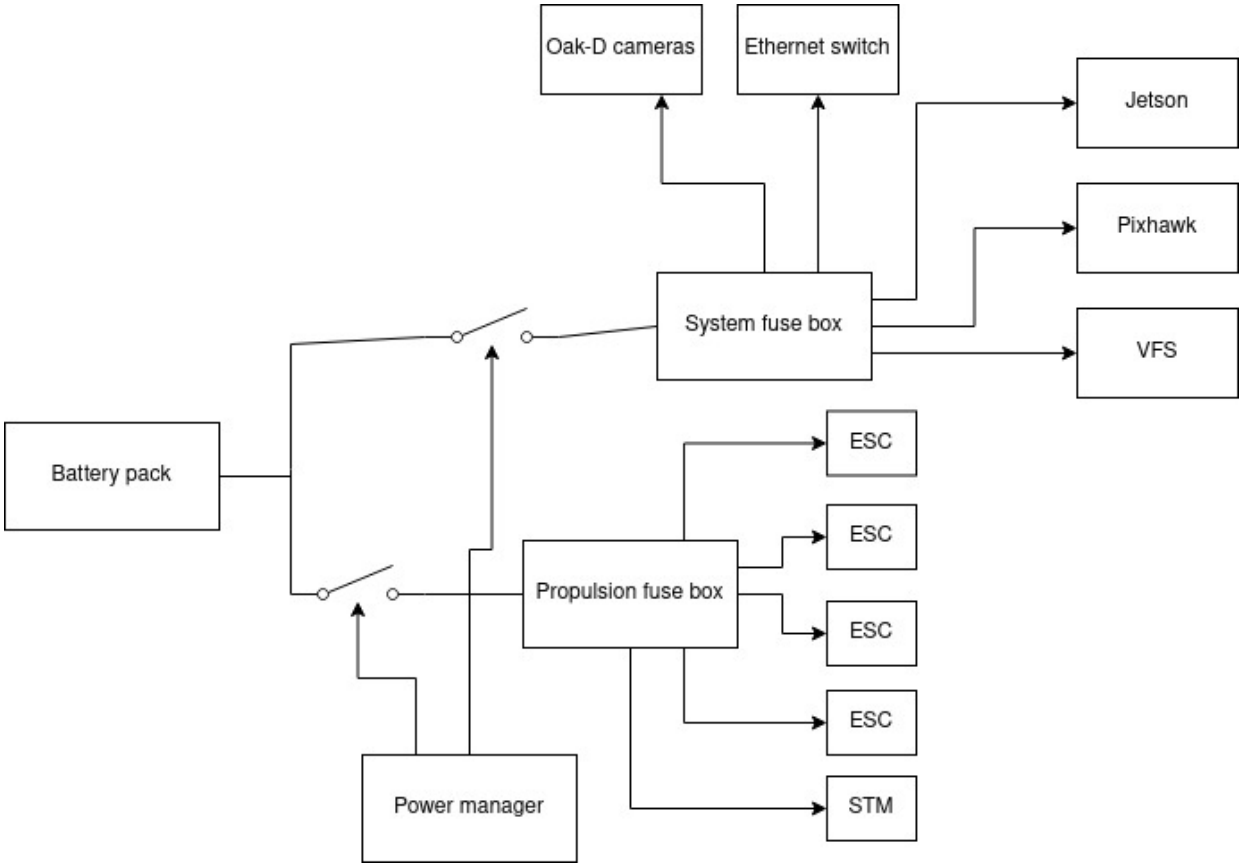
- Image recognition using our new YOLOv8 model works acceptably, it detects all the buoys except the black ones. But frame rate is too slow, and distance estimation is wrong when buoys are too far.
- We are able to control the movement of the ASV from ROS, but our software needs improvement since there are still edge cases causing unpredictable behaviour like spinning.
- When executing waypoint missions in autonomous mode, ASV oscillates from side to side when it is supposed to go in a straight line, which means that our PID controller gains need adjustment.
- Our Wi-Fi communication link and manual control RC link are working without a problem, never dropping the connection at a distance of approximately 30 meters.
- When ASV is in autonomous mode it is only possible to revert to manual mode from the operators' computer not from the RC controller. It means that if we lose our telemetry connection we can not regain control of our vessel unless the telemetry link is re-established. It needs to be improved before next water testing after Christmas.

APPENDIX C
SYSTEM DETAILS

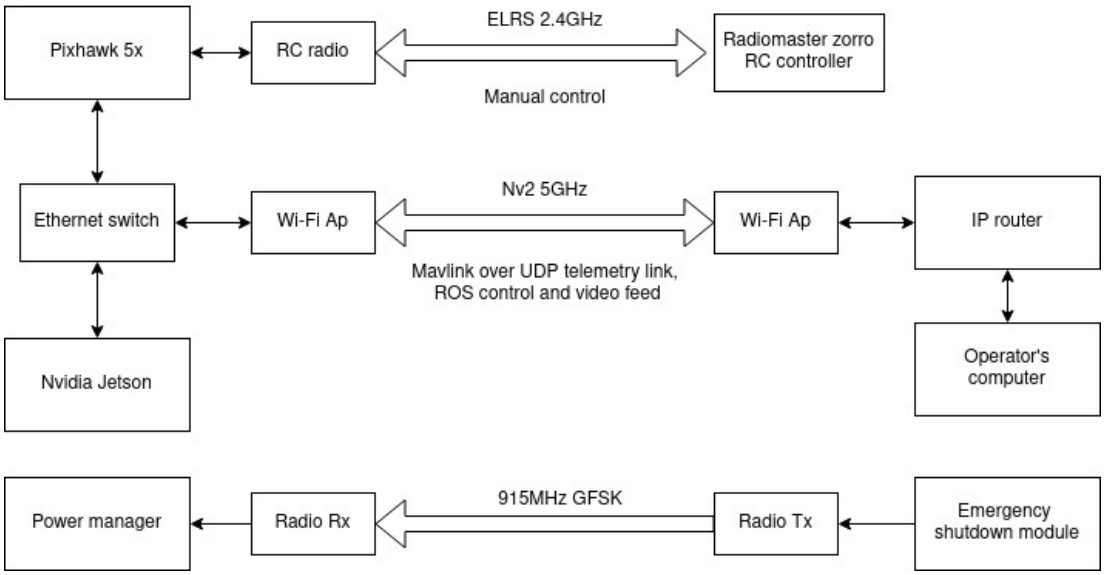
A. Work Breakdown Structure



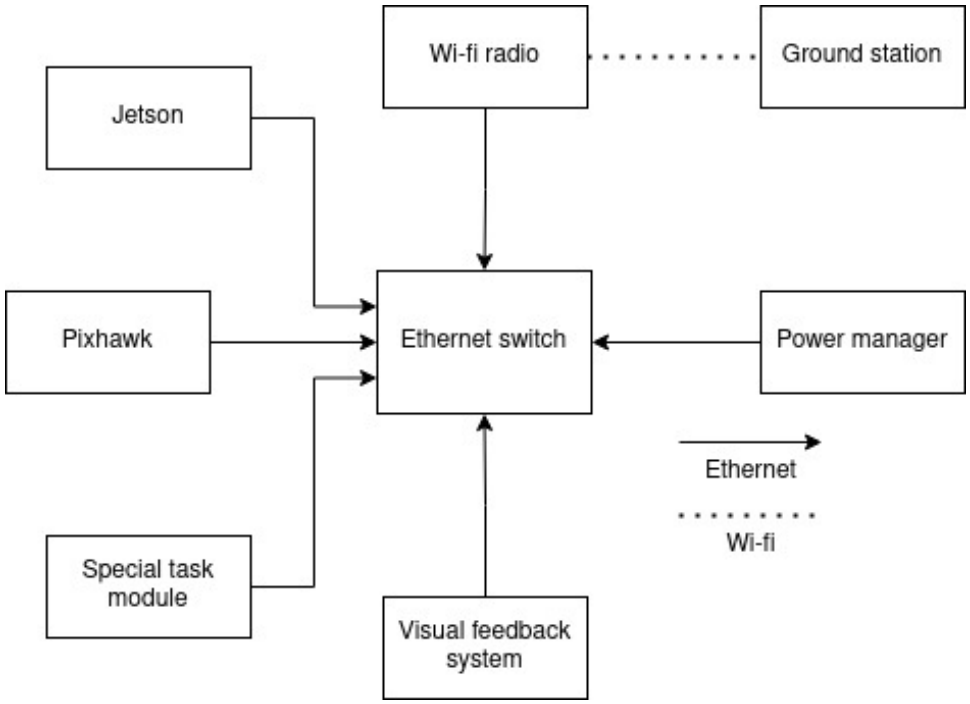
B. Power distribution system



C. Telemetry system



D. Onboard network



E. Thrusters configuration