

D.5.2.2. Swarm of autonomous vehicles with full documentation ready for variety of end-users' applications

InnovaMare project

Blue technology - Developing innovative technologies for sustainability of Adriatic Sea

WP5 – Cooperation in innovation on robotic and sensors solution (TT) – pilot actions

Project References

Call for proposal 2019 Strategic – InnovaMare

Project number: 10248782

Work package: WP5 Cooperation in innovation on robotic and sensors solution (TT) – pilot actions

Activity title: A2 Pilot I. - creating a prototype that is innovative robotic solution

Deliverable title: D.5.2.2. Swarm of autonomous vehicles with full documentation ready for variety of end-users' applications

Expected date: 28/02/2022

Deliverable description D.5.2.2. This deliverable presents the specifications of the autonomous vehicles chosen in INNOVAMARE project to address the monitoring of different environmental variables accordingly with some of the use cases presented in D 5.2.1.

Partner responsible for the deliverable: [PP9-FER]

Dissemination level: CO - Confidential

Status: Final

Version: V1

Date: 28/06/2022

Contents

INTRODUCTION	3
SWARM OF AUTONOMOUS VEHICLES WITH FULL DOCUMENTATION READY FOR VARIETY OF END-USERS' APPLICATIONS	3
Blueye Pro ROV	3
Hardware	3
Software	4
Korkyra ASV	4
Characteristics	4
Hardware	5
Software	11
Possible applications (see table on scenarios of D 5.2.1 below)	13
Mode of operation/Standard Operation Procedure	14
SWAMP ASV	18
Characteristics	18
Possible applications (see table on scenarios of D 5.2.1 below)	23
Mode of operation/Standard Operation Procedure	23
GEOMAR Surface vessel	24
Characteristics	24
Possible applications (see table on scenarios of D 5.2.1 below)	25
Mode of operation/Standard Operation Procedure	25
USE CASE SCENARIOS declared in 5.2.1 to use for reference	Error! Bookmark not defined.
CONCLUSIONS	25
REFERENCES.....	25

INTRODUCTION

The Deliverable 5.2.2 introduces the fleet of autonomous vehicles available in the INNOVAMARE partners and built/operated by INNOVAMARE partners. These include a series of commercial and in-house built autonomous vehicles and support vessels. The deliverable introduces the vehicles, their hardware, main characteristics and mode of operation.

SWARM OF AUTONOMOUS VEHICLES WITH FULL DOCUMENTATION READY FOR VARIETY OF END-USERS' APPLICATIONS

Blueye Pro ROV

Hardware

The Blueye Pro ROV (see Fig. 1) is manufactured by the Norwegian company Blueye Robotics. Its dimensions are 485x257x354 mm (LxWxH). It weighs 9 kg in the air, is designed for a depth of up to 300 m (with a 400 m tether) and has an autonomy of 2 hours provided by its 96 Wh battery. It can be easily trimmed for different underwater environments, be it saltwater, brackish water or freshwater. The ROV has a total of four 350 W thrusters. Two rear thrusters in the horizontal plane, one in the vertical plane and a side thruster allow the vehicle to be highly manoeuvrable along with automatic heading and depth control modes. It can reach a top speed of up to 3 knots and operate in water currents of 2 knots maximum.

The main sensor is a HD camera with a $[-30^\circ, 30^\circ]$ tilt angle mechanism that operates at 25 - 30 fps. Blueye PRO ROV features powerful 3300 lum lights with 90 CRI LEDs that provide well-lit images and excellent colour reproduction. Other sensors include an IMU with 3-axis gyro and 3-axis accelerometer, depth sensor, magnetometer (compass), temperature (indoor and outdoor) and an internal pressure sensor. Additional payload can be attached to the top and/or bottom. The ROV comes with a surface unit that allows the ROV to be connected and controlled via WiFi as well as via Ethernet in challenging wireless environments. The ROV is connected to the ASV Korkyra via an Ethernet cable. Thus, communication between the operator and the ROV is done using ASV Korkyra's NUC main computer as a communication relay.



Fig. 1 Blueeye Pro ROV.

Software

The Blueeye Software Development Kit (SDK), based on Python3, was integrated into ROS2 to allow the operator to monitor the situation underwater while viewing the video stream in the ROS2 graphical user interface (GUI). The ROV-ROS2 GUI displays the ROV battery charge level and enables/disables the automatic depth and automatic heading diving modes. It also switches between manual mode (control via a gamepad connected to the operator's computer) and autonomous mode (in which the NUC main computer aboard ASV Korkyra sends controls based on images recorded by the ROV). The ROV's camera parameters (exposure, white balance, hue, bit rate, frame rate, and resolution), as well as lighting, water density, and boost gain, can be easily changed through this GUI, as shown in Fig. 2. All parameters can be easily changed using buttons, sliders, or combo boxes that prevent any of the parameters from being set outside of their range. The code for the Blueeye SDK-ROS2 interface and GUI is publicly available at [1].

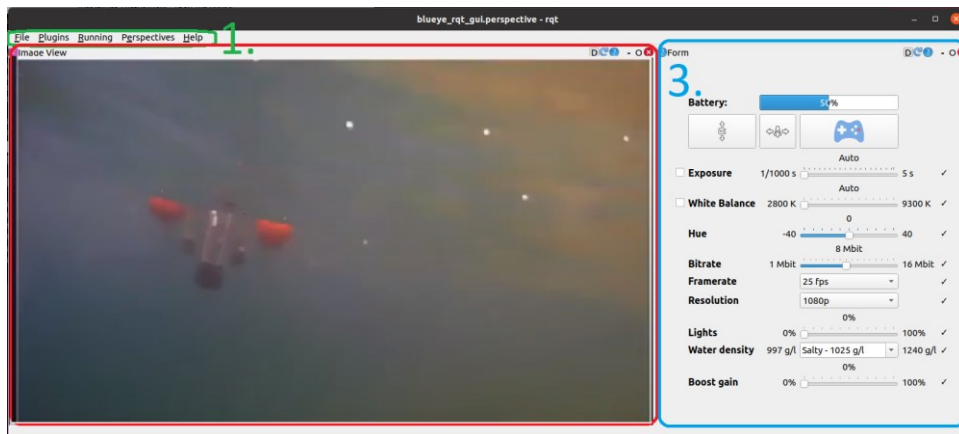


Fig. 2 Graphical user interface of Blueeye-ROS2 interface.

Korkyra ASV Characteristics

Hardware

The autonomous surface vehicle (ASV) named Korkyra (see Fig. 3) is developed as part of the INNOVAMARE project. It is designed as a catamaran to provide better stability and hydrodynamic properties in sea states up to state two. It is made of aluminium, 2000 mm long, 1000 mm wide, and has hollow hulls with a diameter of 240 mm. It has a modular design. The upper deck consists of a carbon hull that houses all the electronics and computers that enable the autonomy of the vehicle.

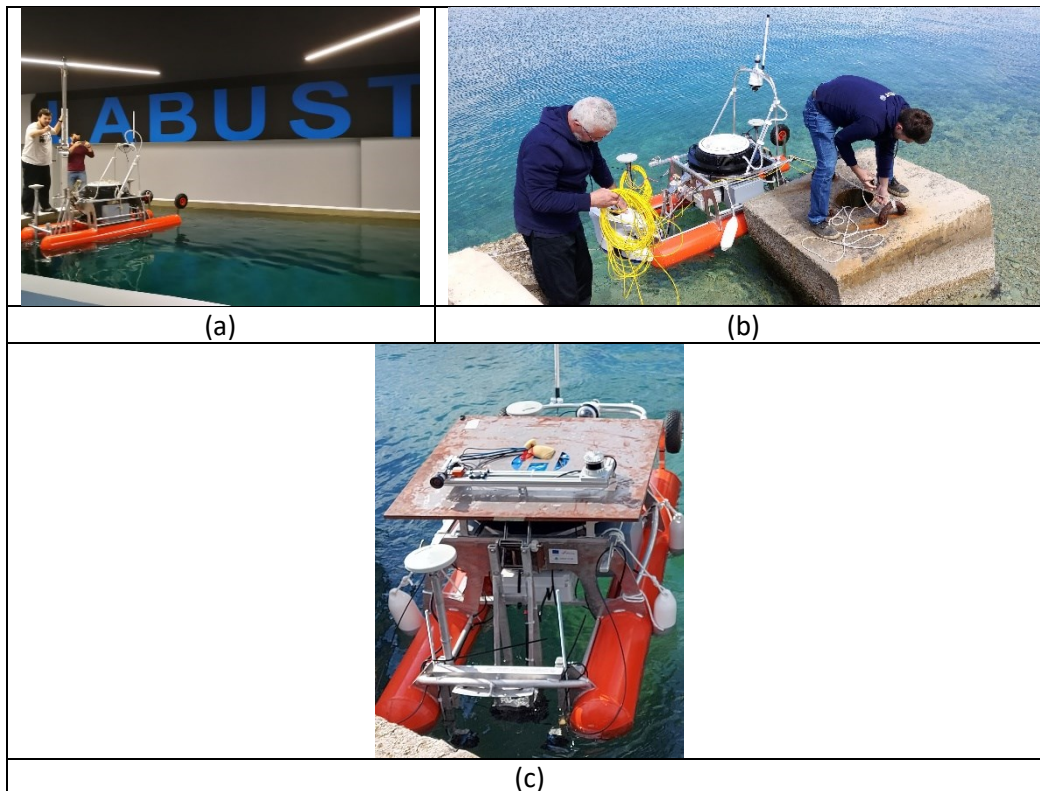


Fig. 3 (a) Testing of the ASV Korkyra at LABUST's pool. (b) Maiden voyage and testing of ASV and ROV controls system integration at sea in Split, Croatia in April 2021. (c) Ouster OS1-128 lidar and FLIR Blackfly S USB camera integration testing at sea in Biograd na Moru, Croatia in October 2021.

The catamaran's lower deck houses IP67-rated watertight aluminium boxes that house additional batteries, motor electronics, the NORBIT iWBMSc multibeam sonar system INS, and an expansion box that provides easy plug-and-play power, Ethernet and USB connectivity for each payload, see Fig. 4. The sonar, which is used in bathymetric applications of the ASV, is mounted in the front of the ASV on a bracket that can be lowered or raised as needed. The roll bar above the upper deck allows integration of maritime signal lights, surveillance camera(s), possibly even LiDAR, etc. It can be easily folded down if a landing platform or other payload is to be mounted above the carbon hull. The design of the carbon hull and the electronics it contains are based on the aPad vehicle previously developed at the Laboratory for Underwater Systems and Technologies (LABUST) [2,3]. The catamaran is 800 mm high from the bottom of the hull to the top of the carbon hull and 1400 mm high to the top of the roll bar. It currently weighs 100 kg in the air. Mobility of the ASV on land and during deployment from shore is provided by two rugged wheels on the rear of the hulls and two swivel wheels on the front. When the ASV is deployed, the rear wheels can be easily lifted and fixed with pins, while the front wheel can be easily removed. The ASV's stability and manoeuvrability at

sea is further enhanced by 300 mm high keels at the front of each hull. On land, these keels act as front legs on which the ASV stands, and they also protect the payload mounted beneath the ASV.

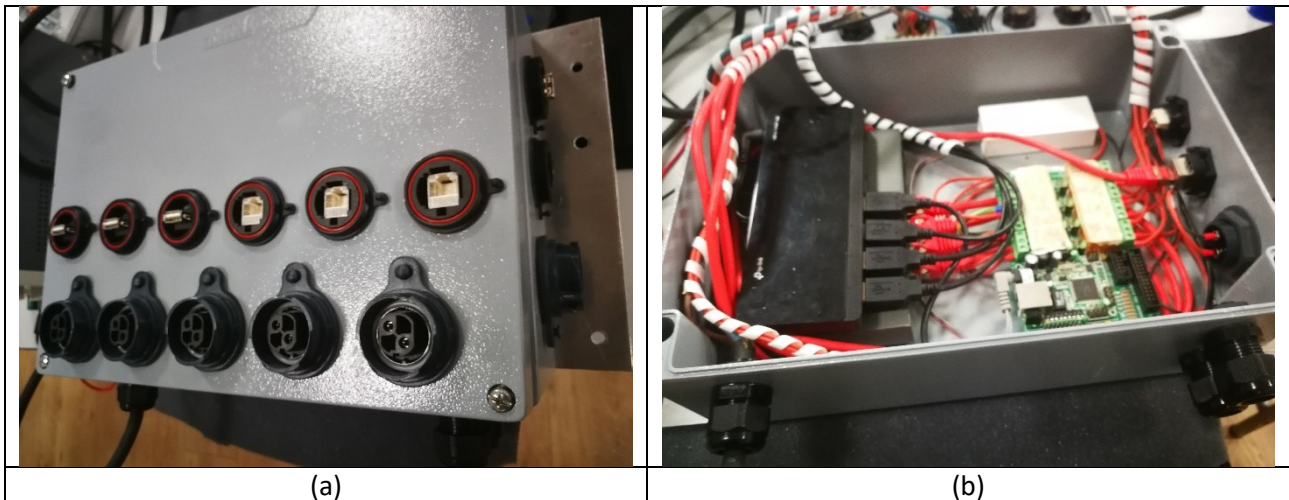


Fig. 4 (a) Interior of the subsystem extension box. (b) Exterior of the extension box.

ASV Korkyra can accommodate an additional 50 kg (in the air) of payload so that the hulls are semi-submerged for best hydrodynamic characteristics. If needed, it can be mounted with a maximum 100 kg (in the air) of payload when its hulls are fully submerged, which would give the catamaran slightly worse hydrodynamic characteristics. Four 390 W electric T200 thrusters in X configuration enable it to navigate complex marine environments at lower speeds (1-2 knots) in all directions. The thrusters are mounted on movable masts that can be adjusted in height. Thruster orientation in the horizontal plane can also be adjusted with a resolution of 45°. In addition, the 720 W Minn Kota RT 55 EM booster electric motor enables top speeds of 3-4 knots. The booster motor is mounted in the rear of the ASV on the same height-adjustable bracket as the sonar.

The total energy the ASV can draw from its batteries is equivalent to 14 BlueRobotics 14.8 V, 18 Ah batteries, for a total of 252 Ah or 3.73 kWh. Worst-case calculations were performed to obtain a lower bound on the autonomy of the vehicle. Assuming that all subsystems of the ASV operate at maximum power all the time, the system power is about 1 kW. If the Minn Kota booster motor is not used at all, which is a realistic scenario in the fish net pen environment where slow and highly precise movements are required, the ASV consumes at most about 200 W. Furthermore, this means that the autonomy of the ASV Korkyra is between 3.5 h and 20 h. In mixed consumption scenarios, where the ASV needs to use a high-power motor to get from shore to fishery quickly and perform slow inspection manoeuvres in all directions, the ASV has an average autonomy of 10-11 h. This is more than enough for the use cases of the INNOVAMARE project.

The GNSS with inertial navigation system IMU, named Applanix SurfMaster, combined with base station corrections over the Long-Term Evolution (LTE) network, enables the ASV to localize itself globally with an accuracy of up to 10 cm, which is very important in survey and inspection missions. The autonomous behavior of the vehicle is supported by two Intel NUC mini PCs. One 7th-generation NUC is used to collect and process multibeam sonar data under Windows, while another 10th-generation NUC under Linux and ROS is used for mid- and high-level control, data processing and mission control. Communication with the vehicle is via Wi-Fi (Ubiquity Bullet M2 and an omnidirectional antenna on both the ASV and operator side) with a peak transmission speed of 100

Mbps over 400-500 m range. The option with Ubiquity Rocket M2 and Ubiquity 120° sector antenna is also being tested to achieve longer range and better bandwidth.

Remote monitoring of the catamaran via the video stream from a pan-tilt-zoom (PTZ) Hikvision IP camera (mounted on top of the roll bar, see Fig. 3) ensures that the operator can respond to any risky situations that might occur at sea. The IP camera is also integrated with ROS, allowing the operator to receive visual feedback from the camera via Wi-Fi.

Acoustics subsystem

Control of the ROV by the ASV needs a position feedback. In the underwater environment this in most cases means using some of the acoustical localization systems, since electromagnetic waves are strongly attenuated. Since the ASV Korkyra's dimensions in the horizontal plane are 2x1m, the use of range-based short baseline (SBL) system with transducers placed at four corners of the ASV (or even wider) is a better choice than an ultra-short baseline (USBL) system. With larger baseline, the underwater localization system is less sensitive to range measurement noise. Furthermore, since Blueye Pro ROV is a small ROV without guest ports to integrate additional sensors, a standalone lightweight battery powered bottomside unit is needed. WaterLinked Underwater GPS (UWGPS) G2 SBL system (see Fig. 5) with a bottom side Locator U1 system met all these requirements.

The UWGPS G2 system is quite convenient since the topside comes in a Pelican case that is easy to integrate onto the ASV. It has an acoustical range of 100m, which is enough for the use case scenarios. However, the range can be upgraded to 300m just by a software update. It's omnidirectional, with an option to change both the maximum range and bearing angle sector to improve the positioning accuracy. UWGPS G2 system comes with an onboard computer that takes GNSS, IMU, and acoustics measurements and reprojects them into the relative position of the ROV w.r.t. the ASV and in georeferenced coordinates as well. There is also an option to provide the topside with external GPS and IMU measurements either by NMEA messages or as HTTP requests to the local HTTP servers which runs all the computations and communication. Ping rate is 2-4Hz.

2

It offers Ethernet and WiFi communication interfaces. Ethernet is better suited for autonomous missions since it has wider bandwidth and less delay than WiFi, and does not interfere with the WiFi that the vehicles are communicating over. WiFi option is however suitable in manual dataset collection tasks or any tasks which do not demand that the ASV is present.

Also, using GPS and heading measurements from UWGPS G2 all the datasets can be georeferenced. Setting all the parameters of the system is made easy by a web GUI that runs on topside's computer, which is accessible through any web browser. Also, this web GUI enables the user to visually track the ROV with its estimated global coordinates in a georeferenced map, as shown in Fig. 5.

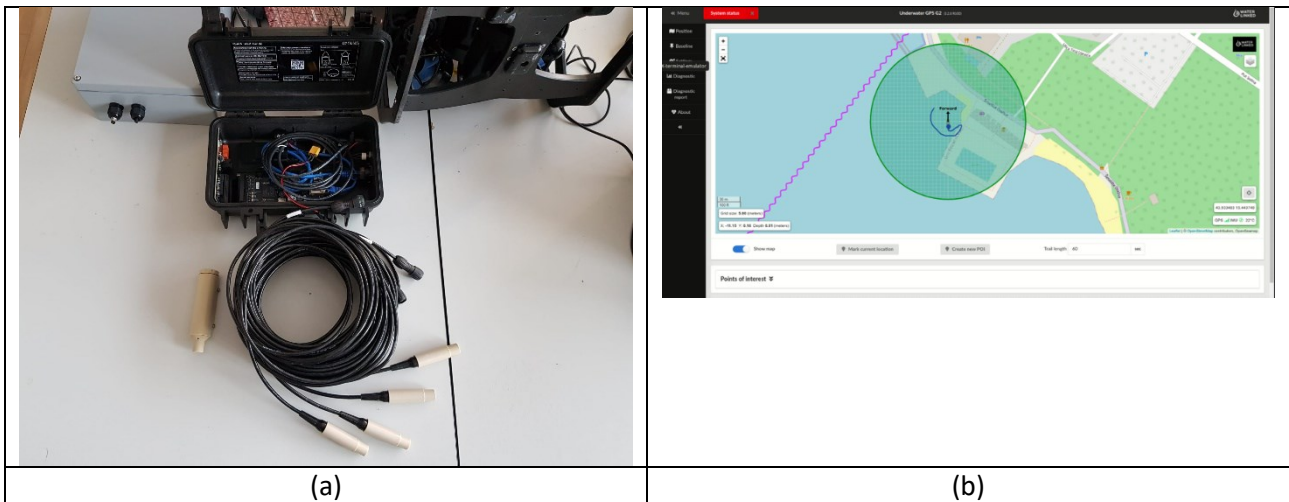


Fig. 5 (a) WaterLinked Underwater GPS G2 topside with 4 transducers for underwater acoustical localization and Locator U1 (left). (b) Web GUI run on the topside computer that allows system setup and tracking of the ROV visually in a georeferenced map.

Locator U1 is a lightweight (75g in water) battery powered bottom side transponder. Its battery can last for up to 10h and is easily charged (6h) using a USB-C port. It is rated up to 300m. It has a GPS chip integrated for time synchronisation with the topside. Indicator LED signals the following states: GPS signal searching state, GPS fix locked state, and no signal state. The bottom side unit is mounted onto the Blueye ROV by using a bracket sold by WaterLinked company. Frequency channel is chosen by a switch inside the cap of Locator U1. Of course, the same channel has to be chosen in the web GUI of the topside to enable acoustic pinging between the topside and the bottom side.

A new SBL system mount with adjustable transponder depth is under development, as shown in Fig. 6 a. It is designed to provide a 2x2m baseline in more open areas, but also a 2x1m baseline in more congested areas around the fish net pens with many buoys and ropes. The depth of each transponder can be changed separately and can be fixed at up to 1m. Furthermore, a foldable floating version of SBL mount is developed and made from plastic tubes with Styrofoam floats, as shown in Fig. 6 b. This allows testing of the acoustical localization system to be uncoupled from the necessity of using ASV Korkyra.

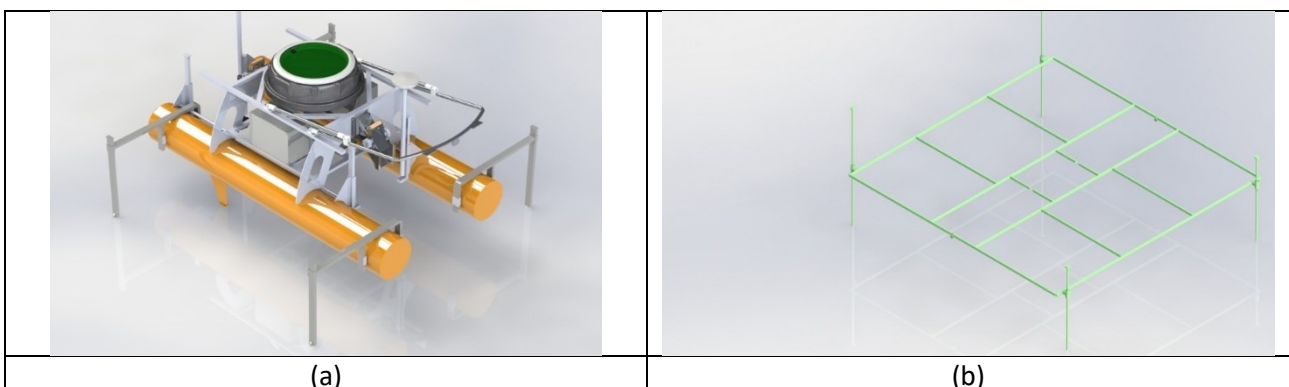


Fig. 6 (a) The finished aluminium acoustics baseline mounting mechanism. (b) 3D model of the foldable floating SBL mount.

Tether management system subsystem

Integration of the ROV onto the ASV and their cooperative path planning in autonomous inspection missions requires a tether management system (TMS) to be developed, as shown in Fig. 7.

The developed TMS model, shown in Fig. 7, is designed as a modular payload to be easily mounted and removed from the ASV Korkyra depending on the use case. It consists of a TMS control box that contains all the electronics needed to run the TMS and control the ROV. Its inputs are connected to the extension box aboard the ASV for power, Ethernet and serial communication. Outputs of the control box are TMS stepper motor power and encoder cables, Blueye cable towards the slipping inside the TMS drum. Once the docking mechanism is developed, the same control box will be used to control it. The motor of the TMS is on the starboard side in a vertical position, connected to the TMS drum over a 90° transmission gear system with transmission factor 1:2 for increased torque. Below the control box is the TMS drum with 400m of wound tether. Precise and uniform (un)winding of the tether is achieved by connecting the two-way spindle to the TMS motor to convert rotational to linear motion. For each complete revolution of the motor, the hook of the linear element through which the tether is pulled through moves one Blueye tether width, i.e. 4mm. Once the hook reaches the edge of the spindle, it automatically starts moving in the opposite direction. The tether then goes over a pulley connected to a spring to keep the steady tension on the tether, and over a second pulley to direct the tether towards the middle of the ASV's hulls during both winding and unwinding.

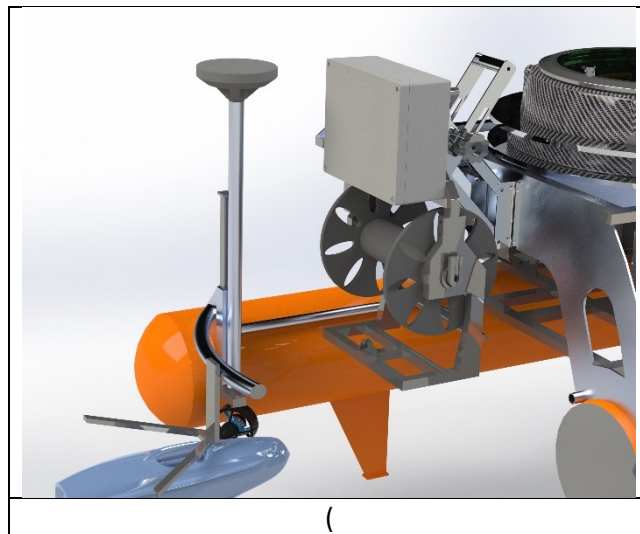


Fig. 7 SolidWorks model of the tether management system and docking station for the ROV aboard the ASV Korkyra. Side view of mounting onto the ASV.

A simple prototype TMS drum mount is developed for preliminary testing of the subsystem at the laboratory pool, as shown in Fig. 8. It uses the same TMS drum, and is connected to the prototype control box. This enables testing of the TMS without having to deploy the whole catamaran into the pool.

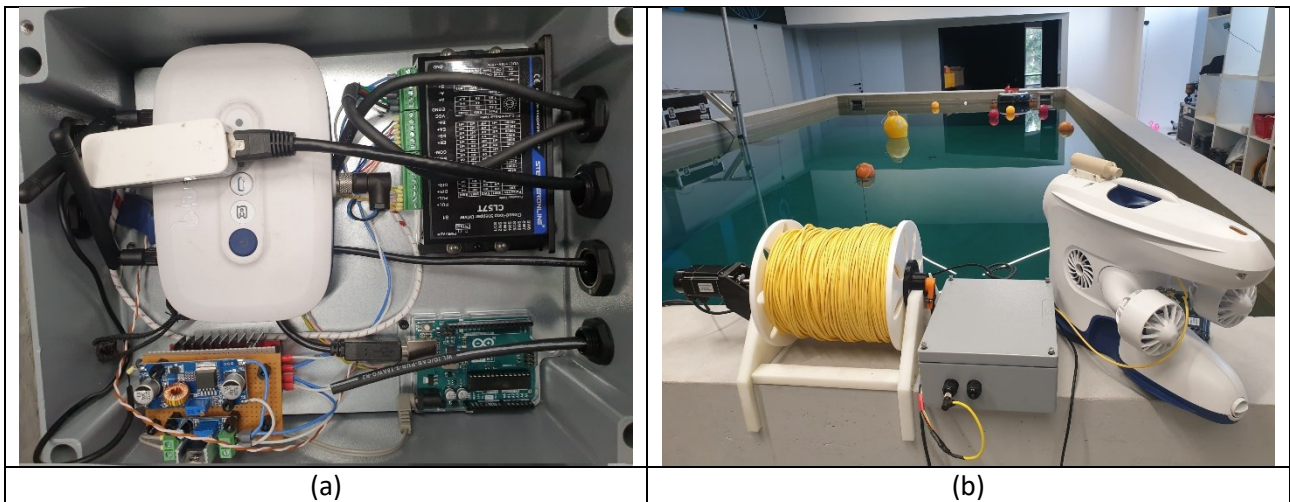


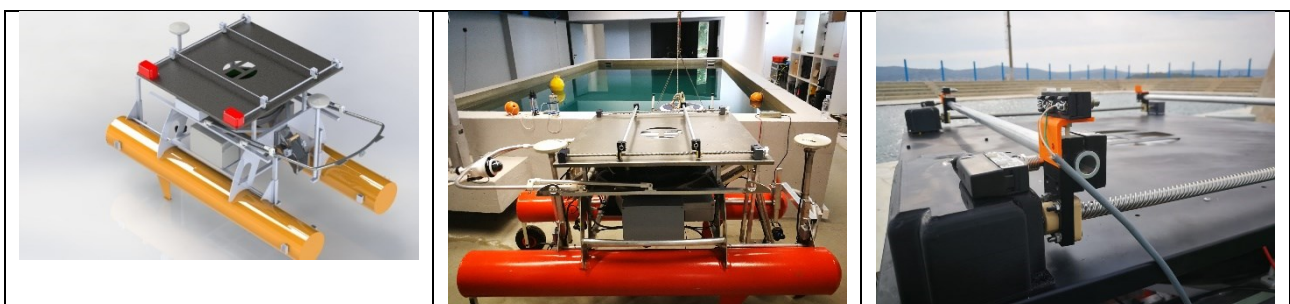
Fig. 8 (a) Aluminium TMS control box for ASV mount. (b) TMS prototype for tests at the laboratory pool.

Landing platform subsystem

As mentioned above, LAAR could perform aerial fish net pen inspections. For this to be done in coordination with the ASV Korkyra, a landing platform (LP) for the LAAR to be docked onto the ASV is developed. Landing procedure is separated into two phases. The first phase includes the LAAR detecting the ASV and the landing platform with a recognizable "H" symbol and activating electromagnets placed at all 4 corners of its landing trap. Once the LAAR is firmly attracted to the landing platform, two strong metal guides mounted onto pulleys and actuated by a step motor will close towards the middle of the landing platform and thus dock the LAAR. After this, the LAAR will deactivate the electromagnets not to cause compass drift.

The model of the landing platform is shown in Fig. 9 a. LP is 1000 mm x 1000x mm x 10 mm (L x W x H) in size. It can be easily mounted on the upper deck of the ASV Korkyra with a 870 mm x 870mm x 200mm (L x W x H) removable frame. The platform is made of a lightweight iron sheet that allows the LAAR's electromagnets to be fixed to the platform during the initial phase of the landing procedure. The iron sheet is covered with a protective layer to prevent it from rusting due to the salty and humid marine environment. The landing platform is connected to the upper deck of the ASV with bolts and nuts so that it can be easily removed from the ASV for missions in which the LAAR is not needed, as shown in Fig. 9 b

LP's control system consists of two Nema 17 bipolar 59Ncm 2A stepper motors connected in parallel to the DM542T digital stepper driver. The control loop is closed over four IP65 rated contact switches (see Fig. 8 c) that detect when the LP is open or closed so that the control algorithm knows when to stop the motors.



(a)	(b)	(c)
-----	-----	-----

Fig. 9 Platform for landing of the LAAR onto ASV Korkyra. (a) SolidWorks model render. (b) Constructed protoversion for lab tests mounted onto the ASV. (c) Stepper motors (in black boxes) and four contact switches integrated for the control system of the LP.

All electronics of the LP are housed in an IP66 rated plastic "control box". This includes stepper motor driver, its DC/DC converter, an Arduino Uno with its 10V voltage stabilizer, and a BlueRobotics 14.8V, 18Ah battery. The box has six cable glands on its side, one for the outside power source, i.e. when it is integrated onto the ASV, one for the USB cable for communication with the main autonomy computer aboard the ASV, and four for the contact switch cables. Power button, motor rotation direction toggle button, and a potentiometer for motor speed control, all used in manual control mode of the LP, are well sealed so that water can't penetrate into the control box.

Software

Korkyra general

Operator work and mission planning for ASV Korkyra will be facilitated by open-source, graphical user interface-based software called Neptus, developed by the Laboratório de Sistemas e Tecnologia Subaquática (LSTS) of the University of Porto. Neptus communicates with the vehicle via Intermodule Communication (IMC) messages, which are simultaneously integrated with ROS via a ROS-IMC bridge.

Camera

The ROS -Qt (RQT) GUI, which includes the video stream and PTZ control, is shown in Fig. 10. In addition to the Real Time Streaming Protocol (RTSP) based video stream, this GUI also allows the operator to control pan, tilt, zoom, and focus and aperture of the camera lens.

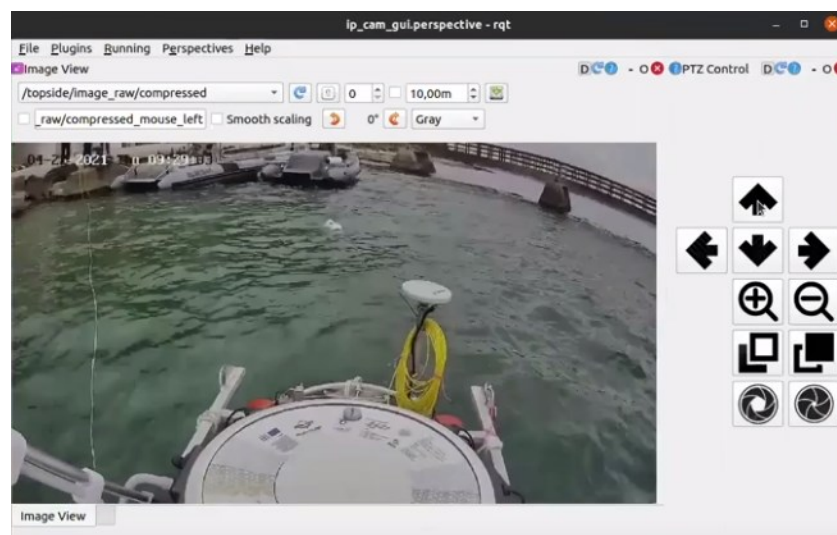


Fig. 10 User interface of the catamaran's surveillance pan-tilt-zoom camera. System integration tests in Split, Croatia in April 2021. Blueeye ROV is visible just in front of the ASV.

Acoustics

A Python v3.9 API developed by the manufacturer enabled easier software integration of the UWGPS G2 system. A ROS2 package was developed to integrate the acoustical localization measurements into the ROS framework. It sends HTTP request to the topside's HTTP server, parses the returned relative and global positions and publishes them to ROS topics. The code of the package is open-source and publicly available at [4].

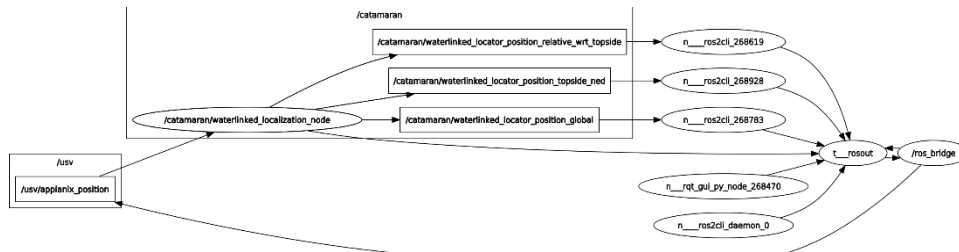


Fig. 11 Node graph of the UWGPS ROS2 interface with ROS1-ROS2 bridge for GNSS and orientation measurements.

TMS

Manual control of the TMS was implemented for easier initial prototyping and testing, and whenever the TMS is not mounted onto the ASV. Auto control mode of the TMS assumes that the commands are sent to the Arduino Uno inside TMS' control box via serial communication. On the topside the TMS is integrated into the ROS1 framework as a serial communication node that uses an TMS Arduino-specific USB port. It sends commands from a certain topic directly to the set serial port, and also publishes incoming messages from that port to another topic. The commands are sent periodically at 2-4Hz which is the upper limit of underwater localization system's position estimation frequency. An example of ROS-TMS integration is shown in Fig. 12 where the commands are sent over WiFi to the ASV whose main autonomy computer communicates over USB interface with the (for now) power-independent TMS control box. The command requests in the above defined format are relayed at the topside from ROS topics onto serial interface, and vice versa.

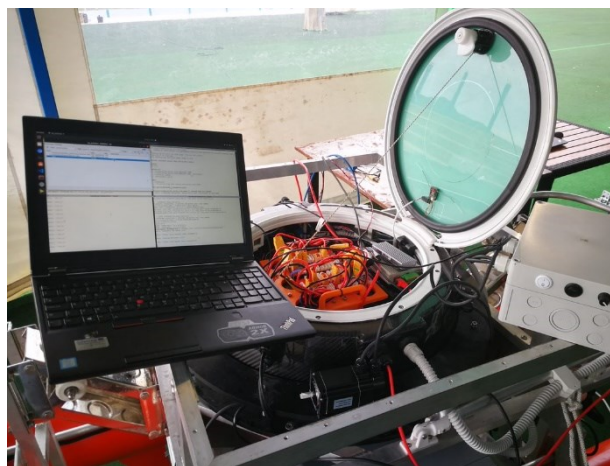


Fig. 12 TMS integration tests in auto mode while the TMS control box is connected to the ASV Korkyra.

LP

Manual control of the LP was implemented on an Arduino Uno for easier initial prototyping and testing, and whenever the LP is not mounted onto the ASV. Four contact switches are used as feedback even in the manual mode, so that the person testing the LP doesn't have to worry about

the guides slamming into the LAAR's landing skies when closing or the motor box when opening. Auto control mode of the LP assumes that the commands are sent to the Arduino Uno inside LP's control box via serial communication. These commands are envisioned as a simple communication protocol in auto control mode between the topside, i.e. the main PC onboard the ASV, and the bottom side being LP's Arduino Uno.

On the topside the LP is integrated into the ROS1 framework as a serial communication node that uses an LP Arduino-specific USB port. It sends commands from a certain topic directly to the set serial port, and also publishes incoming messages from that port to another topic. It is important to note that the LP's Arduino parses new messages only after it completes its previously requested command.

Possible applications (see table on scenarios of D 5.2.1 below)

The ASV Korkyra can be equipped with various payloads, such as a multibeam sonar (MBES), a remotely operated vehicle (ROV), a tether management system (TMS) for the ROV, a landing platform for the LAAR, as conceptually shown in Fig. 13, on the upper and front lower parts of the ASV. The cooperation of the ASV with the ROV and/or the lightweight autonomous aerial robot (LAAR) can be used for inspection and monitoring of aquaculture ecosystems to map and detect biofouling in aquaculture cages (MSFD 5).

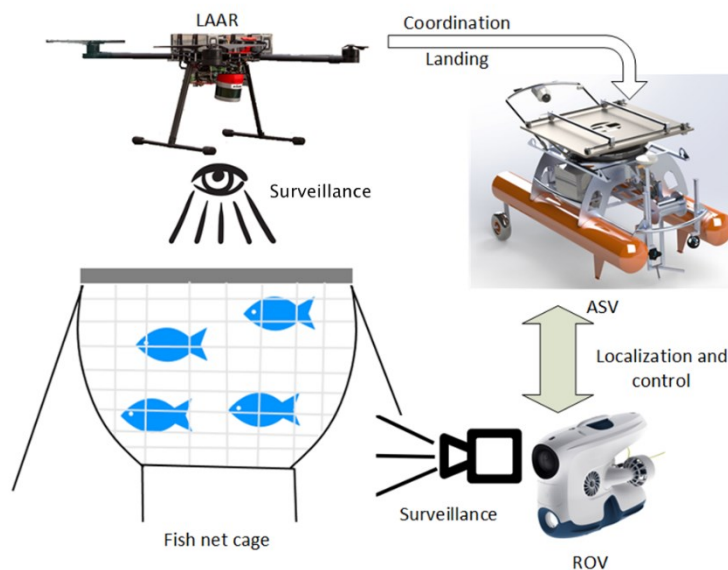


Fig. 13 Conceptual design of autonomous mariculture inspection system

In its configuration with the MBES, the ASV Korkyra can be used for inspection and monitoring of harbours and marinas including marine litter detection. This encompasses creating high resolution bathymetric 3D models using MBES and/or mosaic of the seabed (MSFD 6). Furthermore, it can be used for sediment erosion estimation near man-made structures (MSFD 7), automatic (offline) marine litter detection on mosaics (MSFD 10), as well as creating high resolution digital orthophoto (aerial), surface litter/pollution detection on orthophoto map and creating interactive maps/GIS for charting/tracking pollution/litter.

Mode of operation/Standard Operation Procedure

Getting started

1. ROS-related installations (Linux OS assumed)
 - a. Install ROS1 Noetic, ROS2 Foxy and ROS1-ROS2 bridge on the topside operator's computer.
2. Other installations
 - a. Install Neptus, IMC and DUNE developed by LSTS. Source code from which these applications can be built are available as open-source online. Prebuilt binaries of Neptus Mission control and review software will be provided by the LABUST team.
 - b. Make sure that you have korkyra.navc vehicle description file in neptus/vehicle_defs folder of the root Neptus binary folder. This will enable Neptus to send mission commands to ASV Korkyra once its in Auto mode.
3. Network setup
 - a. Topside/operator computers should be in the available 10.0.10.100-250/16 IP address range for basic ASV Korkyra usage. This holds for Ethernet or WiFi interfaces depending on which one is used.
 - b. It is recommended to connect to the vehicle either directly by a Ethernet cable plugged into its network switch or over only one wireless network interface. This way the congestion of the wireless communication is minimized. There are two options to connect to the vehicle wirelessly:
 - i. Recommended: Connect the topside computer to a switch to which a Wi-Fi with preconfigured Labust-WiFi access point is connected as well. The ASV Korkyra connects to this WiFi SSID by default. Details about Labust-WiFi network setup will be provided by LABUST team if needed.
 - ii. Connect to Labust-WiFi with login details given by the LABUST team.
 - c. An overview of devices connected onboard the ASV Korkyra is given in Table 1. For the default username "stdops" the default UNI_PASS password will be provided by LABUST team.

Element	IP address	Configuration	User	Password
Bullet WiFi bridge	10.0.108.1/16	WDS mode to Base Station	stdops	UNI_PASS
Main computer - NUC10 Ubuntu	10.0.108.2/16 192.168.1.108/24 192.168.53.155/24		stdops	UNI_PASS
Sonar computer - NUC7 Win10	192.168.53.150		stdops	UNI_PASS
NORBIT MBES	192.168.53.100 -> 10.0.108.4?			
Ethernet switch carbon hull				
HikVision IP camera	10.0.108.5/16		stdops	UNI_PASS
WaterLinked UW GPS G2	10.0.108.6/16	WiFi: SSID: UnderwaterGPS Pass: waterlinked @	stdops	UNI_PASS

		192.168.7.1 Eth default: 192.168.2.94		
SNMP 8 Relay Module @ carbon hull	10.0.108.7/16	5-Camera, 6-Light, 7-ExtensionBox	stdops	UNI_PASS
SNMP 8 Relay Module @ extension box	10.0.108.8/16	1-Sonar, 2-UWGPS, 3-LP, 4-TMS, 5-Lidar?	stdops	UNI_PASS

Table 1. Network setup of all devices connected onboard the ASV Korkyra

4. Powering on the ASV Korkyra and its subsystems

- a. ASV: Plug in the power cable from the battery to the power board (into port with PWR label). Remove the magnet to turn the main electronics on.
- b. Main relay board: For energy saving reasons, not all subsystems (if connected) are powered on by default. This is done using IP SNMP 8-relay boards. The main relay board is housed inside the carbon hull of the ASV Korkyra.
 - i. It can be controlled through any browser at 10.0.108.7 address.
 - ii. In "I/O Ports" screen, camera, lights and extension box can be turned on/off by checking the checkbox next to the desired device and then click Change I/O state button. After you click this button, an audible click from the switched relay will be heard from the inside of the carbon hull.
 - iii. Networked devices controlled by the relay boards can be pinged at the above listed IP address to check if they are turned on and working properly.
- c. Auxiliary relay board: The extension box that is used to connect and power MBES sonar, UWGPS acoustics, the landing platform and the tether management system.
 - i. When powered on, it is controlled by any browser at 10.0.108.8 address.
 - ii. Make sure that the power, Ethernet and/or USB cables of the desired payload are connected to the extension box.
 - iii. In "I/O Ports" sonar, UWGPS acoustics, the landing platform and the tether management system can be turned on/off by checking the checkbox next to the desired device and then click Change I/O state button. Alternatively, the extension box has tagged 1-8 power outlets so make sure to turn on the relay at the correct number.
 - iv. After you click this button, an audible click from the switched relay will be heard from the inside of the extension box.

5. Vehicle status overview

- a. In your web browser go to 10.0.108.1 address. If everything is set up correctly, Labust GUI should appear, showing battery status of the vehicle. For Li-ion batteries, they should give 16V initially when not being under much load. Do not let the battery voltage go below 12-13V!

6. Manual control

- a. After a few minutes, turn on the Spektrum joystick, put it into MANUAL mode (lower right switch), and test to see if all the thrusters work correctly.

7. Automatic control

- a. On the topside computer, open Neptus and see if the vehicle is visible in the map, or at least IN SERVICE mode. Normally you should see Korkyra's green icon in the map at the correct location if its GPS module is getting the data, i.e. is not obstructed by vegetation, relief etc.
- b. Plan a mission for ASV Korkyra by clicking on Plan icon on the top left. Add Goto waypoints and Lawnmower patterns with desired velocities.

- c. Save the plan.
 - d. Change operation mode on SPEKTRUM joystick to AUTO.
 - e. Run the plan (icon on the top right in Neptus).
8. ROS-related things for developers
- a. If you wish to access the data that is sent through ROS onboard the ASV Korkyra's main NUC10, you can easily access the vehicle by typing in "ssh stdops@10.0.108.2" secured with UNI_PASS password.
 - b. Since both ROS1 and ROS2 nodes will probably be running on startup, you'll need to source the correct ROS version by "source /opt/ros/noetic/setup.bash" or "source /opt/ros/foxy/setup.bash". Also build/install spaces should be sourced accordingly "source /catkin_ws/devel/setup.bash" for ROS1 Noetic or "source /ros2_ws/install/setup.bash" for ROS2 Foxy.
 - c. After this you can echo, list, get info etc. about any ROS1/2 topic.
 - d. If you wish to use your own ROS packages, you can clone your repos into catkin_ws/src for ROS1 or ros2_ws/src for ROS2 packages, build the code and run/launch the nodes as you wish.

Bathymetric surveys

1. Connect to Korkyra's onboard Sonar NUC7 mini PC via remote desktop application.
2. Start WBMS application. See if you get readings from the sonar. If not, check to see if the sonar is really turned on, or the application needs to wait for some time to establish communication with the sonar.
3. Establish a connection of the NUC mini-PC to Internet either via USB tethering, or a USB mobile internet stick. Connect your phone to the NUC via USB cable, and turn on USB Tethering option. In Windows aboard NUC allow this connection.
4. Open NTrip application from NUC's desktop and type in the information needed for GPS corrections. It should connect and start getting the data if the vehicle is not occluded by trees etc. In case you do not have access to this data skip this step.
5. On the topside computer, open Neptus and see if the vehicle is visible in the map, or at least IN SERVICE mode. Normally you should see Korkyra's green icon in the map at the correct location if its GPS module is getting the data, i.e. is not obstructed by vegetation, relief etc.
6. Put Korkyra down into the water either by a crane or manually.
7. Using Spektrum joystick in MANUAL mode, move the Korkyra away from the launch site, and away from a possibly GPS-denied area.
8. Try commanding Korkyra to move around and rotate in place to check if it is getting GPS and heading data. If it is getting the GPS data, it should move and rotate in Neptus as well.
 - a. If Korkyra is not moving and/or rotating in Neptus:
 - i. Close Neptus.
 - ii. In Console type in "killall java".
 - iii. Start Neptus again.
9. If you already have preplanned missions in Neptus for Korkyra to cover, skip to the next step. Otherwise, plan the mission as advised in Subsection 1.7.
10. Before each trial, it is recommended to run patch tests for pitch, roll, and speed of sound calibrations defined in NORBIT manual page 126-130.
11. Reopen WBMS application over remote desktop and check if the sonar parameters are all set as they should be.

- a. Set Swath parameter to 90deg or less. Going higher than 90-100deg usually produces outliers on the edges of sonar swath due to possible interactions with the floaters of the vehicle itself.
 - b. If examining a structure from its side, set the Angle to 30deg (starboard side up is positive roll angle direction), and swath up to 90-100deg.
 - c. Set Ping rate to Adaptive unless otherwise instructed.
 - d. Set Gate width to Normal unless otherwise instructed.
 - e. Set the frequency for 400-500kHz in deeper waters (20-50m depth), and 700kHz in shallow waters (under 20m depth).
 - f. Check Roll stabilization option.
 - g. Set the Equidistant option.
12. If using HYPACK application for logging into .hsx format, be sure to open HYPACK, and then click on menu Survey-> Hypack Survey and Sweep.
 13. On Spektrum joystick switch to AUTO mode.
 14. In Neptus, right click on the location near the start of the mission and click Command Korkyra-> Go here to move the vehicle towards the area where its mission is planned.
 15. When the vehicle reaches its planned mission operating area, open WBMS and click on the red Record icon to start logging sonar and position data. A big [RECORDING] label should appear.
 16. In Hypack Survey and Sweep application, press Ctrl+S to start logging, or click File-> Start Logging. in case that HYPACK Survey and Sweep crashes:
 - a. Make sure to end logging,
 - b. restart HYPACK Survey and Sweep, and
 - c. Start logging again.
 17. Monitor the battery voltage level during the missions as described in step 5. During sonar operations NIDO Robotics Li-ion batteries were used up to 2 hours until their voltage under the load dropped down to approx. 11.8V, and lead-acid batteries until their voltage dropped to around 11V under the same conditions.
 18. When the vehicle finishes its mission, open WBMS and again click on the red Record icon to stop logging sonar and position data. Accordingly, if using HYPACK for logging, reopen Hypack Survey and Sweep application, and press Ctrl+E to end logging, or click File-> End Logging.
 19. Repeat steps 14-18 to cover as much area as possible, and as long as the batteries hold. When the battery voltage becomes too low, you will notice that in WBMS the connection to the sonar will be lost meaning that the sonar shut down due to too low voltage for it to operate.
 - a. In case that the battery voltage gets very low, command Korkyra to come close to your operating area/boat as described in step 17. If for some reason you lose connection to Korkyra in Neptus, try bringing it back by Spektrum in MANUAL mode. If even this fails, retrieve Korkyra by a boat.
 - b. Retrieve Korkyra back to the operator's area/boat and change the batteries.
 - c. Repeat steps 1-18.
 20. After finishing for the day and/or in case of low disk space on Korkyra's NUC mini PC:
 - a. Plug a LAN cable in Korkyra's network switch to connect your computer.
 - b. Connect to Korkyra's NUC via a remote desktop.
 - c. Copy all (WBMS and/or HYPACK) logs needed from NUC to the shared folder on your computer, as described in Subsection 1.5.4. There should be shortcuts to "WBMS logs" and "HYPACK logs" folders on NUC Desktop.
 - d. In case of low disk space on NUC, delete the logs after making sure that they've been successfully copied to your computer.

Net pen inspection missions

1. ROS-related

- a. Clone and build all needed Github repositories for controlling ASV camera, acoustics, TMS, LP or other subsystem. Access to these repositories will be provided by LABUST team. Instructions about dependencies of these packages are given in README file of each package.
 - i. Camera streaming and PTZ control repo: https://github.com/labust/hikvision_ros
 - ii. Topside operator GUI repo: <https://github.com/labust/pladypos-topside-pkg>
 - iii. Blueye SDK-ROS2 interface and GUI: <https://github.com/labust/blueye-ros2-pkg>
 - iv. UWGPS acoustics ROS2 pkg: <https://github.com/labust/waterlinked-uwgpsg2-ros2-pkg>
 - v. Landing platform control ROS1 pkg: https://github.com/labust/hektor-ros-pkg/tree/main/landing_platform
 - vi. Tether management system control ROS1 pkg: https://github.com/labust/hektor-ros-pkg/tree/main/tether_management_system

2. Main launch files are described for each package in their respective README files.

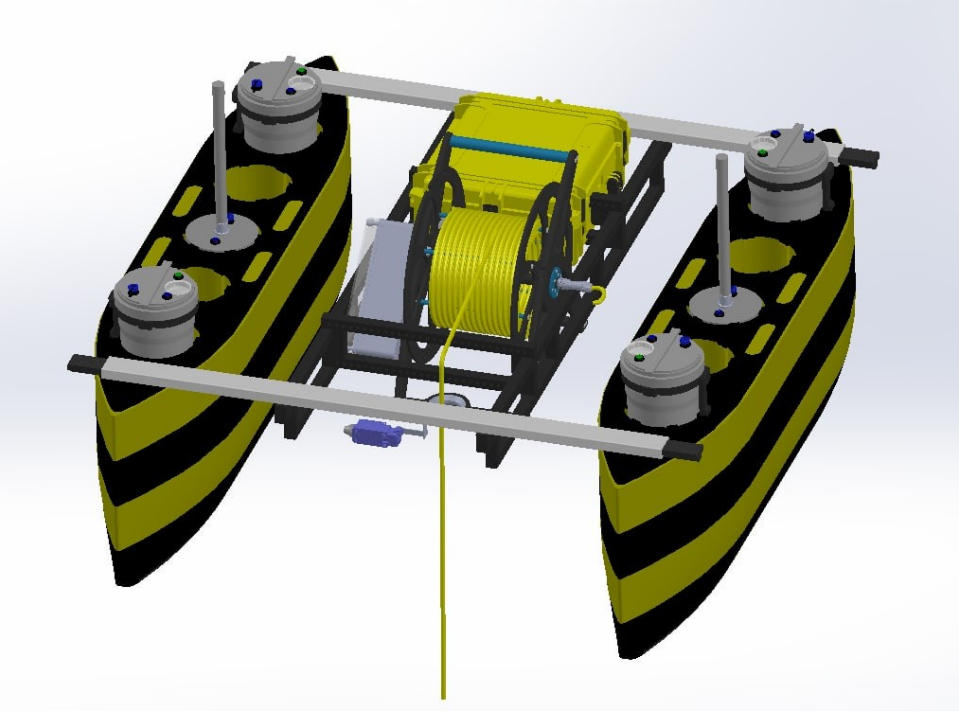
SWAMP ASV

Characteristics

SWAMP [5] - Shallow Water Autonomous Multipurpose Platform - is a portable, modular Unmanned Surface Vehicle (USV), designed and built by CNR-INM Genoa research group. It is a catamaran, equipped with four azimuth Pump-Jet thrusters, all contained within the hulls, and designed specifically for SWAMP. The hulls are made of a soft-foam lightweight material, each hull hosts a propelling and control unit - MINION -.

The design of SWAMP ensures high modularity and floatability, rendering the vehicle adaptable to the mission and user requirements. The physical modularity is reproduced also in the software architecture. Moreover, the hulls are fully independent, each of them includes its control, guidance, power, propulsion, navigation and communication systems. SWAMP has an onboard Wi-Fi communication network, that enables communication among every single element, other than between the two hulls. This design renders SWAMP a completely modular vehicle that can be dismounted and transported, and then remounted in various possible configurations.

SWAMP ASV allows for extremely shallow water navigation, disclosing unprecedented data in a hostile navigation environment.



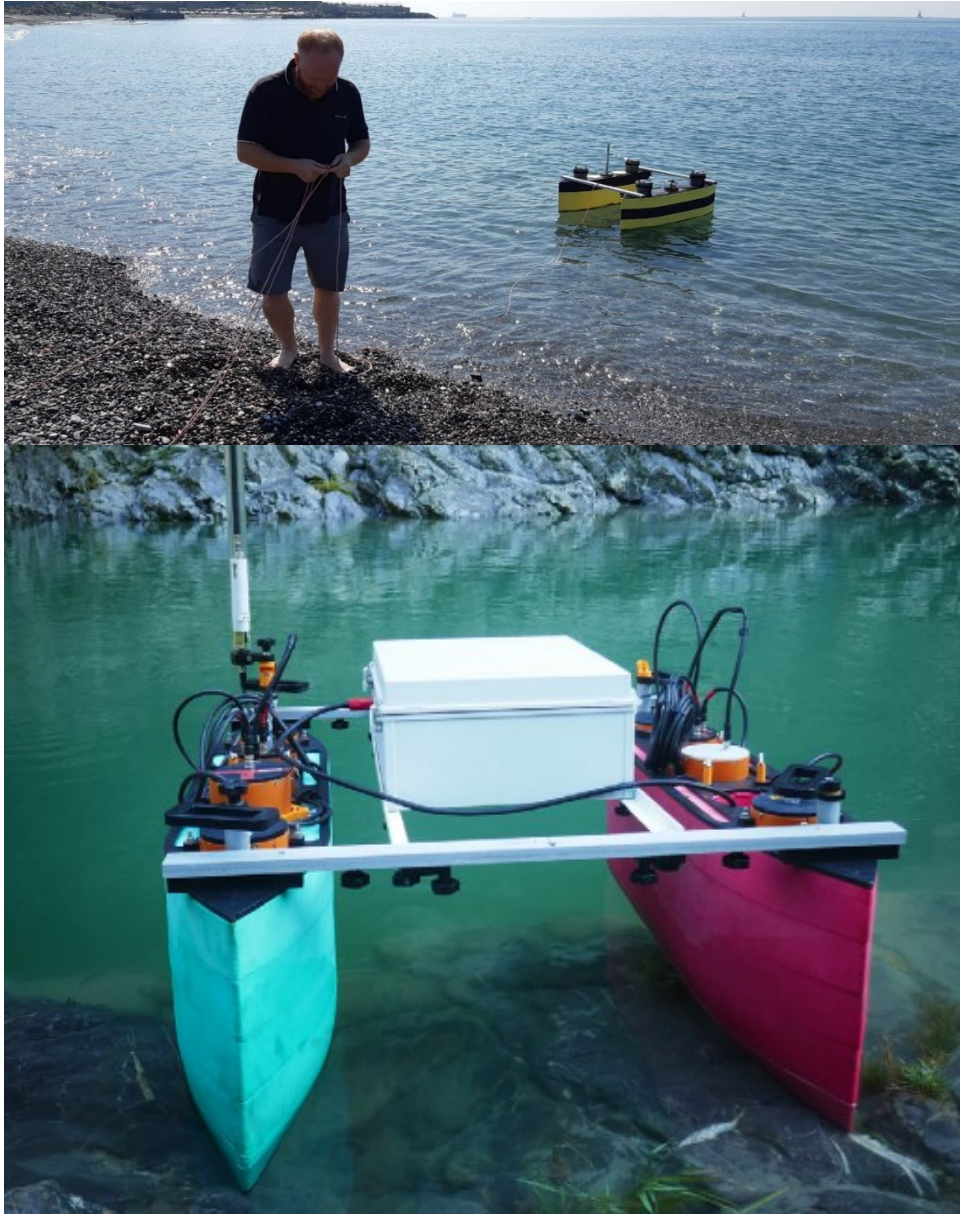


Figure 14: SWAMP in a river (bottom) and sea (top) deployment

Hardware

SWAMP is a 1.23 m long fully electric catamaran with a design width of 1.1 m; the width is variable between 0.7 m and 1.25 m. The hull height is 0.4 m; the whole vehicle, including the antennas and the structure connecting the two hulls, is approximately 1m high. SWAMP unloaded and dry weight is 38 kg with a draught of 0.1 m. The standard maximum payload is 20 kg, resulting in a maximum design draft of 0.14 m. SWAMP's buoyancy reserve allows for load up to 60 kg with a draught of 0.22 m. The shape of the hull is inspired by the Wigley series; it is symmetrical bow-stern and it has a flat bottom. The longitudinal buoyancy centre is located in the middle of the hull. The following figure shows the main dimensions of the Catamaran.

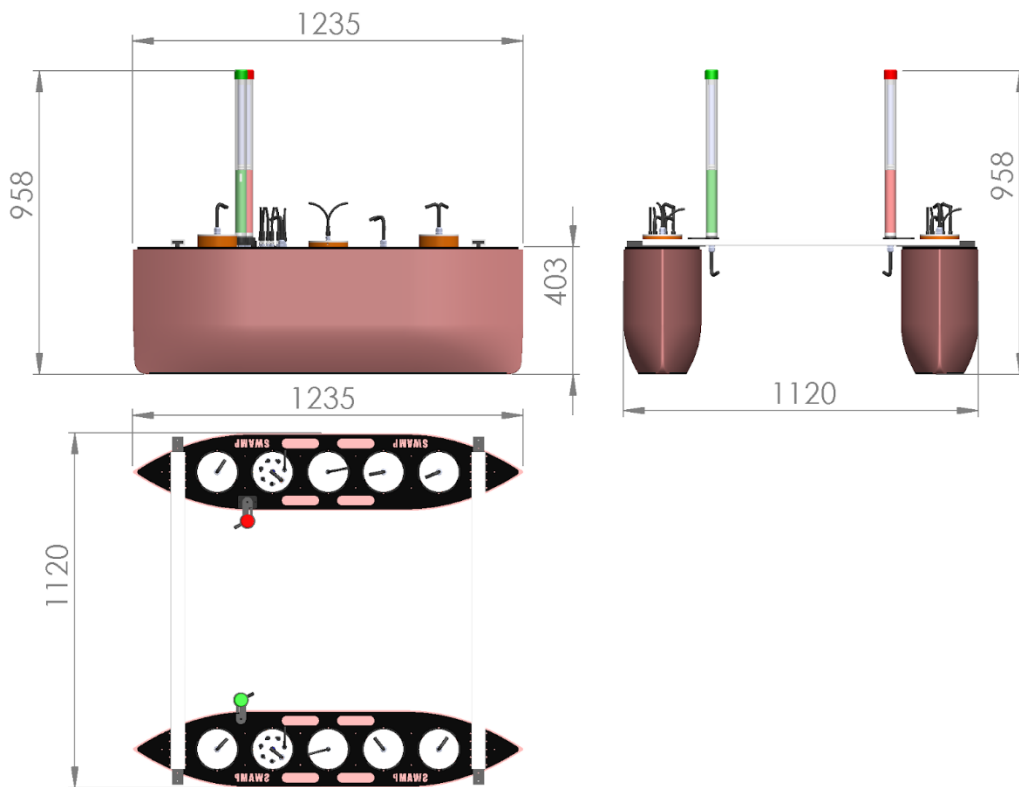


Figure 15: SWAMP main dimensions.

SWAMP is made of a lightweight, soft and flexible structure. The structure is made of layers of soft PE foam with closed cells, HDPE plates and tie-rods and pultruded rods. This design makes of SWAMP a completely unsinkable vehicle.

The maximum speed of SWAMP in deep water is 1.45 m/s, while the speed in extremely shallow water, up to 200 mm, is reduced to 0.9 m/s due to the peculiar hydrodynamic effects that occur in shallow water.

SWAMP is equipped with an intelligent winch kit with a communication cable for the management of underwater sensors and tools. It also equipped with A GPS-RTK kit for highly accurate positioning in the range of centimeters.

In the newest version of SWAMP there is the least possible wiring, which has been reduced to just the power connections made with IP69k connectors and wiring.

Four azimuth thrusters give SWAMP the controllability that is required for high quality surveys. The Propelling and Control Units - MINION - encompass two actuators per monohull based on azimuth thrusters pump-Jet and with brushless motors, their controllers, control and power electronics.

Each Pumpjet Thruster produces 12.25 N at 1200 RPM and a Power consumption of 90 W.

Maximum speed of SWAMP in infinite depth waters is 1.6 m/s, while the speed in extremely shallow waters down to 200 mm is reduced to 1 m/s due to the peculiar hydrodynamic effects occurring in shallow water.

The propulsion modules are illustrated below.

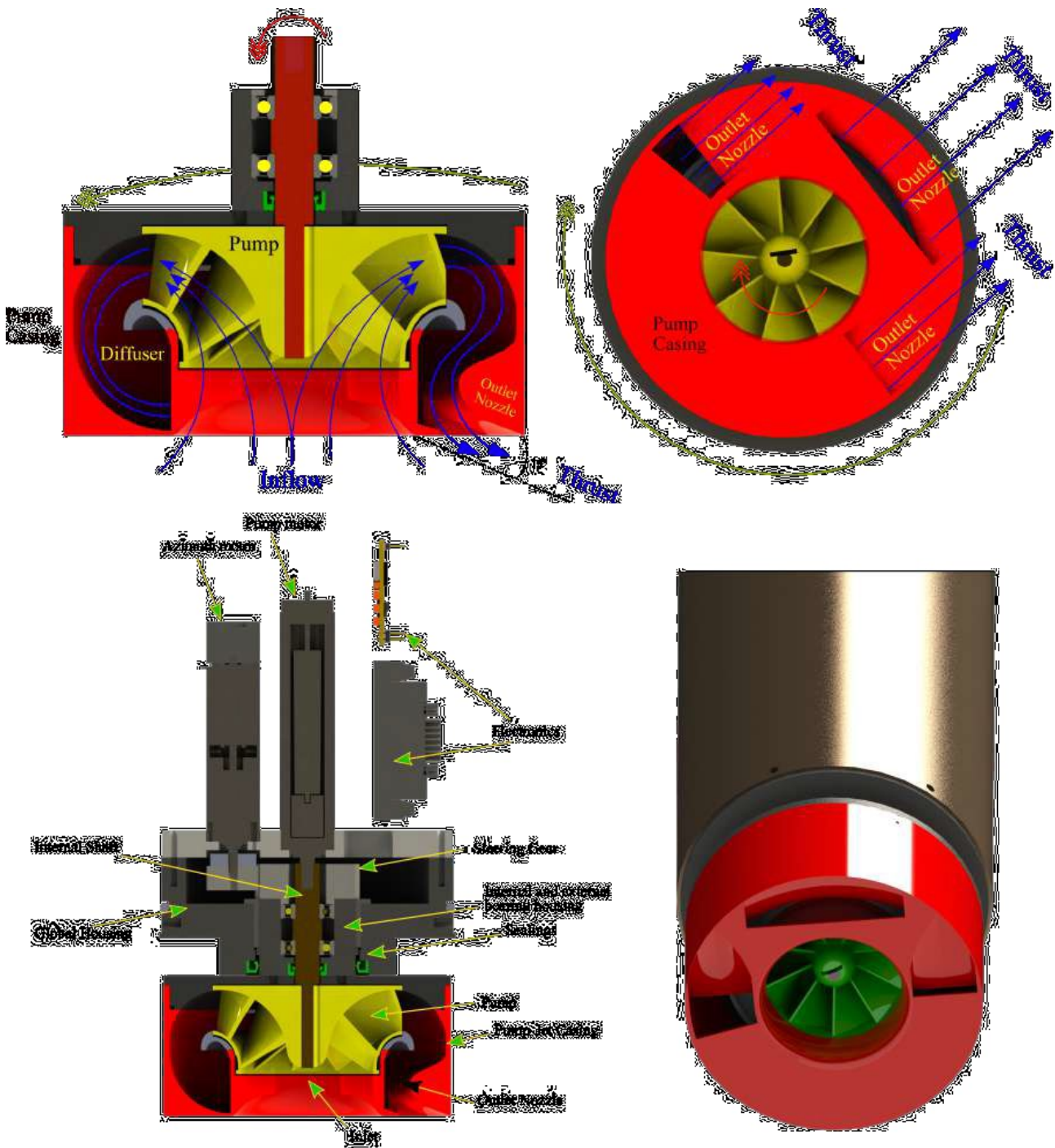


Figure 16: Pump-jet schematics

In the new version of SWAMP the propulsion unit has an optimised temperature management. The innovative motor-propeller coupling is magnetic, hence shaft free.

The battery is integrated in the MINION, it is rechargeable and each propulsion unit has its own battery.

Each MINION is equipped with an electronic interface PCB, which enables control and renders each minion an independent computer. Under this aspect the minion could be an independent vehicle. Inside the MINION there is a GPS and a IMU to ensure localisation and motion performance monitoring.

The Wi-Fi module (i.e. on board access point) mirrors the independency and modularity that figures in the mechanicals parts of the MINION. In fact, the WI-FI module has its own battery and can be used to communicate with the particular MINION, moreover it also allows for connection with and among the other modules.

Software

SWAMP's software architecture is based on standard commercial components.

The basic NGC package of each hull consists of an IMU and a GPS.

The communication module for each hull which provides a communication facility for both its own hull and the other hull's modules. The communication module is a Pilot station Wi-Fi radio (with Omnidirectional antennas for pilot station and Vehicle Wi-Fi radio that can work both in AP (Access Point) mode and in SA (Station Adapter). When the two hulls are physically connected, one of the two computing modules can become the master and the other one can become the slave.

The computing module is based on a Raspberry Pi 3.0 SBC, model B, with Raspbian OS operating system. The software of the control system of each individual hull can be loaded and executed on the computing module. Inside the case of the computational module there are also the basic navigation sensors constituted by an IMU and a GPS.

The basic idea is to have a vehicle made up of N modules (control modules, actuators, sensors, etc.) that communicate via a Wi-Fi connection using the 192.168.29.0 network.

Command packets sent to AM modules are ASCII strings derived from the C programming language. The commands are sent over a datagram network connection.

All the commands developed for the SWAMP architecture can be used from a computer connected to the network. Commands can be sent from a C program, Matlab, Simulink or other programming tools in a simple way.

Preliminary testing

SWAMP will be tested in the surroundings of CNR-ISMAR headquarters in Venice during the week 23-27 May. During the tests a division of the CNR-INM research group will perform SWAMP ASV identification, navigation, guidance and control tests. A second phase of these tests will focus on sensor integration and data communication, in collaboration with other partners of the project.

Possible applications (see table on scenarios of D 5.2.1 below)

SWAMP's modular configuration allows a wide range of diverse application. Depending on the payload of the vehicle, it can perform:

- Shallow/deep waters environmental monitoring
- Bathymetric surveys
- River patrolling
- Water and air quality monitoring in urban serving damns

Mode of operation/Standard Operation Procedure

SWAMP is controlled remotely by a human operator, who connects to the onboard access point and guides the vehicle.

GEOMAR Surface vessel

Characteristics

The INNOVAMARE partner GEOMAR has a Surface Vessel (SV) that can be used in support of trials. This surface vessel can be used for various tasks such as Multi-beam sonar (MBES) surveys, ROV tasks and sampling by various sensors. This vessel (of up to 10m) can serve a platform for performing various tasks, surveys as well as an office at sea, especially in locations such as ports, mariculture farms, etc. The current vessel can be seen in Fig. 26.



Fig. 17 GEOMAR's hydrographic vessel (8 m).

Hardware modules

The surface vessel has the same modularity as the SWAMP ASV and the Korkyra ASV. Typical sensors mounted in this boat are:

- multibeam echosounder (MBES) Teledyne RESON SeaBat T20-P;
- sound velocity profiler Valeport mini SVP;
- Trimble-Applanix positioning system;
- geological parametric sub bottom profiler (SBP) Innomar ses 2000 Light for geological surveys.

ROVs such as the SIRIO Ageotech in can also be launched from the Surface Vessel and tracked using an Ultra-Short BaseLine (USBL) Trittech system. For more info on the sensors, please consult D5.2.3.

Software modules

For the surface vessel, a captain pilots the boat and therefore there are no autonomous functions. However, a set of proprietary software is used to process the data collected by the sensors. An example of a bathymetric map produced by the software on-board the SV is presented in Fig. 27.

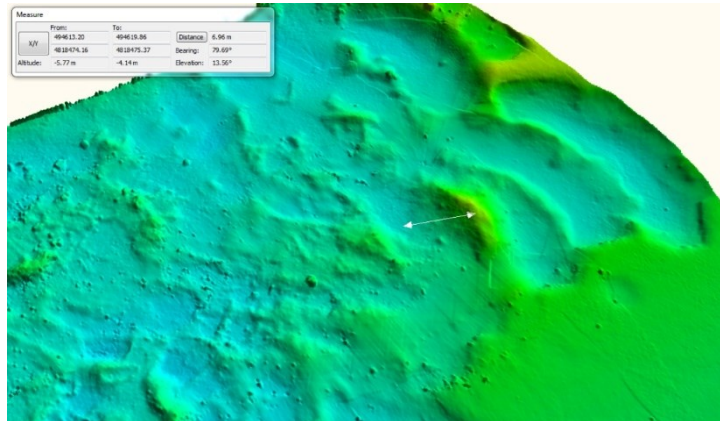


Fig. 18 3D model of the harbour seabed (after dredging)

Possible applications (see table on scenarios of D 5.2.1 below)

- Inspection and monitoring of aquaculture ecosystems
- Inspection and monitoring for harbours and marinas including marine litter detection

Mode of operation/Standard Operation Procedure

As mentioned, the vessel is piloted by a captain while different commercial software is used depending on the configuration and sensors mounted for a specific application.

CONCLUSIONS

The Deliverable 5.2.2 described the swarm of autonomous vehicles available in the INNOVAMARE partners and built/operated by INNOVAMARE partners. Besides describing the vehicles, the deliverable included mode of operation with instructions for end-users. The vehicles are available for end-users in a series of scenarios (detailed in D5.2.1) but can be adapted to other purposes. These vehicles will be used in the Demonstration scenario 2 as described in Deliverable 5.2.5.

REFERENCES

- [1] Kapetanović, N. and Vuković, J. (2021). BlueyeSDK-ROS2 interface. <https://github.com/labust/blueye-ros2-pkg.git>. [Online; accessed 22-July-2021].
- [2] Nađ, Đ., Mišković, N. and Mandić, F., "Navigation, guidance and control of an overactuated marine surface vehicle", Annual Reviews in Control, vol. 40, pp. 172–181, 2015.
- [3] Lončar, I. Babić, A., Arbanas, B., Vasiljević, G., Petrović, T., Bogdan, S. and Mišković, N., "A heterogeneous robotic swarm for long-term monitoring of marine environments," Applied Sciences, vol. 9, no. 7, 2019
- [4] Kapetanović, N. WaterLinked Underwater GPS G2 Python API-ROS2 Interface. 2021. Available online: <https://github.com/labust/waterlinked-uwgpsg2-ros2-pkg> (accessed on 5 November 2021).

[5] Odetti, A., Bruzzone, G., Altosole, M., Viviani, M., Caccia, M., “SWAMP, an Autonomous Surface Vehicle expressly designed for extremely shallow waters”, Ocean Engineering 216 108205, 2020
<https://doi.org/10.1016/j.oceaneng.2020.108205>