RAMONES
Radioactivity Monitoring in Ocean Ecosystems

Deliverable

Document ID: D2.2 Report on Systems Integration and Validation

31/03/2023

RAMONES funded by European Union under Horizon 2020 FET Proactive programme via grant agreement No. 101017808.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Document Info

Project Information			
Acronym	RAMONES		
Name	Radioactivity Monitoring in Ocean Ecosystems		
Start Date	1 Jan 2021	End Date	31 Dec 2024
Program	H2020-EU.1.2.2. - FET Proactive		
Call ID	H2020-FETPROACT-2020-2	Topic	FETPROACT-EIC-08-2020 - Environmental Intelligence
Grant No	101017808	Instrument	RIA
Document Information			
Document Id	D2.2		
Document Title	Report on Systems Integration and Validation		
Due Date	31-Dec-2022	Delivery Date	31-Mar-2023
Lead Beneficiary	IST-ID (2)		
Beneficiaries (part.)	NKUA(1), IST-ID (2), EVOL(3), NTUA(5), PLOA(6)		
Editor(s)	David Cabecinhas (IST-ID)		
Authors (s)	Antonio Pascoal, David Cabecinhas, Pedro Batista, Luis Sebastiao, (IST-ID), Emanuele Coccolo, Sergey Yakovlev (EVOL), Konstantinos Karantzalos, Valsamis Ntouskos (NTUA), Angelos Mallios (PLOA), Javier Escartin (ENS), Konstantinos Nikolopoulos (UDUR), Lydia Maigne (UCA), Theodoros Mertzimekis, Varvara Lagaki, Ioannis Madesis, Georgios Siltzovalis, Polytimos Vasileiou (NKUA)		
Contributor (s)	Eleni Petra, Effie Zafirakopoulou, Stavroula Kazana, Konstantina Bejelou (NKUA)		
Reviewer(s)	Angelos Mallios (PLOA)		
Workpackages	WP2 - Autonomous Marine Robotics for Collaborative Radioactivity Mapping		
Version	v1.0	Stage	Submission
Distribution	Public		
Keywords	Pathfinder, EIC, FET, Radioactivity, Environmental Intelligence, Underwater Research	Type	Report

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Document Change Record

Version	Date	Change Description	Editor	Change Location (page/section)
0.9	17/03/2023	Initial draft version	David Cabecinhas (IST-ID)	
1.0	31/03/2023	Submission version	David Cabecinhas (IST-ID)	

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Disclaimer

RAMONES is a European Innovation Council (EIC) FET Proactive project in the Environmental Intelligence Scope B, related to radically novel approaches to resilient, reliable and environmentally responsible in-situ monitoring, funded by European Union under Horizon 2020 FET proactive programme, via grant agreement No. 101017808.

RAMONES project's main objective is to close the current marine radioactivity under-sampling gap and foster new interdisciplinary research in ocean ecosystems. RAMONES will invest a significant effort to provide tools to enable long-term data acquisition missions, rapid deployments, low cost per information byte, and propose new AI and Robotics-driven and supported methodologies, being ambitious to eventually offer scaled-up solutions to researchers, policy makers and communities. All these may be achieved by combining state-of-the-art (SoA) methodologies and equipment from various disciplines in a well-balanced synergy, and designing new and effective methodologies targeting the marine environment, which will provide efficient response to existing natural and man-made hazards, and shape future policies for the global population. RAMONES will additionally contribute to shaping a blueprint on Environmental Intelligence in the EU and worldwide.

This document has been produced through funding from the European Commission. The content of this document is a product of the RAMONES project Consortium and it does not necessarily reflect the opinion of the European Commission. The editor, author, contributors and reviewers of this document have taken any available measure in order for its content to be accurate and lawful. However, neither the project consortium as a whole, nor the individual partners that have implicitly or explicitly participated in the creation and publication of this document may be held responsible for any damage, financial or other loss or any other issue that may arise as a result of using the content of this document or any of the project outputs that this document may refer to.

The European Union (EU) was established in accordance with the Treaty on the European Union (Maastricht). There are currently 27 member states of the European Union. It is based on the European Communities and the member states' cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice, and the Court of Auditors (http://europa.eu.int/).

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Table of Contents

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Tables of Figures & Tables

Document Figures

Document Tables

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Abstract

A network of tightly integrated computational systems and vehicle assets is required to bring to fruition the innovative RAMONES concept of autonomous adaptive radioactivity monitoring in ocean ecosystems. For the envisioned RAMONES missions, the assets to consider are the fixed benthic station, the two underwater gliders, the autonomous surface vehicle, acting as the main communications hub in the network, and a mission control station, located either on shore or on a crewed surface vessel. Each of the assets will be equipped with a low-power computational unit running the software algorithms for the vehicle's navigation and control, the inter-asset communications, data processing and data-flow management, and logging of the radioactivity sensor measurements.

The underwater vehicle's native rudimentary navigation solution is based on best-guess estimates for the vehicle's velocity and the resulting position is of insufficient accuracy for the RAMONES goals of precise radioactivity mapping and source localization. Furthermore, in typical glider operations it is not possible to adapt mission parameters based on sensor measurements as the mission needs to be defined *a priori*. The RAMONES proposed solution is to integrate an advanced positioning system based on ultra-short baseline (USBL) acoustic devices, providing accurate position measurements, and to exploit the *backseat driver* option of the gliders to enable mid-mission control goal updates.

The USBL acoustic devices double as modems for communication of arbitrary data. By sharing radioactivity environmental signals, the vehicles are able to cooperate in a concerted manner for extended mapping range and can react to measurement events in order to improve the radioactivity source localization. The RAMONES system employs reusable communication modules throughout the assets that abstract the acoustic modem interface and parse the USBL positioning messages. These are validated in realistic simulations to ensure the safe and reliable operation of all robotic vehicles before any experimental tests.

Finally, an early prototype of the integrated RAMONES system has been developed to validate the proposed approach before all assets are available. The RAMONES software builds on existing work by IST-ID and Evologics and has been used to perform cooperative missions with surface and underwater robots, with acoustic communications, in conditions akin to those expected for the final RAMONES demonstrations. The RAMONES system architecture was validated with promising results for a successful final integration once the robotics systems are complete and fully functional.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

1. Introduction

This report presents the current state of the ongoing RAMONES work on the integration of the Navigation, Motion Planning and Control, Positioning and Communication systems in a cohesive, flexible, and easily extensible software system and its respective validation using state of the art hardware-in-the-loop (HITL) simulations and experimental field tests.

The proposed software system will be a crucial enabler of the RAMONES vision of monitoring Ocean radiation through the cooperation of two Autonomous Underwater Gliders (AUG), an Autonomous Surface Vehicle (ASV), and a Benthic Station, towards the critical tasks of underwater localization and relaying of different data streams to other vehicles, as well as benthic and shore stations.

The RAMONES software system builds on the foundational Medusa and NetMarSys software stacks, the former running on the actual vehicles and the latter in simulation, developed at IST-ID throughout years of operation and simulation of cooperative heterogeneous marine robots. Within WP2, these software packages were refactored to simplify the simulation of multiple vehicles (which can now be done using a single computer). The guiding principles of the refactor were simplicity of use and configuration, modularity of the ROS nodes, and extensibility of the integrated system. This software overhaul is a fundamental steppingstone for the development of the integrated RAMONES system.

The newly gained modularity and extensibility of the RAMONES system are exploited for the seamless integration of Evologics' software emulator of the acoustic network, allowing RAMONES partners to quickly plan and conduct realistic simulation tests involving all the software system components. Having hardware-in-the-loop simulations that use the same application programming interfaces (APIs) and exercise the same computer code pathways as the real hardware in the vehicles ensures a swift and smooth passage from simulation to experimental tests, with the confidence that all software runs on the target vehicles and performs as expected. Moreover, the new system expedited the inclusion of a new silent interrogation scheme for the modem-USBLs, enabled by the chip-scale atomic clocks (CSACs) integrated in the modem-USBLs, thereby reducing active signaling and minimizing energy consumption.

1.1. Medusa and NetMarSys

The Medusa [1] and NetMARSys [2] toolchains, developed by IST-ID, denote the complete software stack used for mission management, vehicle control and navigation, visualization, and communication. It encompasses the code running on the actual vehicles and allows for seamless Hardware-in-the-Loop (HITL) and Software-in-the-Loop (SITL) simulations and real-time mission visualization.

The software stack, updated within RAMONES, makes use of the modern Robot Operating System (ROS) Noetic framework, running on the Ubuntu 20.04 LTS operating system, and is written in a mix of C++ and modern Python 3 code. Earlier Medusa versions were developed for running on the older Ubuntu 18.04 operating system and used ROS

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Melodic, which only supports python 2.7, a now unsupported language set as “end-of-life” since January 2020.

Each vehicle (or vehicle simulation) runs its own navigation, motion planning, and control modules – a set of core packages internally known as FAROL (Free Autonomous Robots for Observations and Labelling). Communication among vehicles can be achieved through Wi-Fi, at the surface, or using acoustic modems, in an underwater setting. The network topology is flexible and specified through configuration files stating which vehicles are allowed to communicate and over which channels.

The vehicle network mission is monitored through a shore console that receives vehicle state information and displays on a webpage a regional map overlayed with the localization of each vehicle, together with useful state information. This webpage can also be used to issue mission commands to the robotic vehicle fleet. The mission control software is named PONTE, standing for Page for Operation, Navigation and Tinker Engagement.

A faster development pace is achieved if instead of actual vehicles the software can be developed and tested using simulation models of the marine vehicles. With a simple change of configuration options, the same FAROL packages can run against the actual vehicle hardware or against one of two options for vehicle dynamics simulation. The first is a simple dynamic model node, coded in C++ for speed and efficiency, whereas the second is a fully-fledged physics simulation using Gazebo with the UUV Simulator addon [3], affording higher fidelity and a realistic simulation of vehicle’s movement, sensors, and actuators. The Gazebo simulator was chosen as it is tightly integrated with the ROS framework, in which FAROL is also developed.

Simulation of nonlinear acoustic propagation effects was implemented during RAMONES through the integration of the Software Emulator of Acoustic Network of Evologics within the Medusa framework.

With the code evolution and consolidation brought forward by the RAMONES project, the Medusa and NetMARSys software stacks were open-sourced and made available to the marine robotics community to freely use, build up, and expand. A complete simulation environment with several robotic vehicles such as Medusa, BlueROV 2, and others, is available at https://github.com/dsor-isr/dsor_simulation and related repositories.

1.2. Modem emulator

Evologics is providing the RAMONES consortium with underwater acoustic (UWA) modems for a scalable solution for distribution of data streams among different underwater platforms and localization by means of ultrashort baseline (USBL) “synchronous” uni-directional transmissions, bi-directional, or “silent” localization through triangulation using measurements of the USBL antenna. The modem-USBL equipment is typically accessed using the S2C intelligent Navigation and Positioning Software (SiNAPS) which requires human intervention.

To attain the RAMONES autonomy goals, a more useful software interface is DMAC, a ROS driver for Evologic’s modem-USBLs that allows other ROS nodes to send and receive messages through the acoustic channel, as well as relative positioning messages from other modem-USBL, typically in pairs of range and bearing/elevation angles.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Paralleling the situation for the robotic vehicles, faster turn-around times are achieved if the software to interface the modem-USBL equipment is developed and tested with a stand-in emulator, in controlled conditions, instead of having to deploy the actual equipment in the ocean. Evologics has then made available to the RAMONES consortium not only the acoustic hardware but also a cloud solution for the emulation of all features of the modem's data-link protocol layer and including a phenomenological simulator of the physical acoustic channel. Parameters such as modem position, orientation, frequency, source and signal levels, attenuation, error rates, among others, can be set arbitrarily and exploited to emulate a variety of ocean conditions without having to leave the laboratory. This simulator is fundamental to test the robustness of the RAMONES system against physical phenomena in long range and varying depth acoustic channels that are not reproducible in controlled environment tests.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

2. The integrated RAMONES system

2.1. RAMONES concept

The main objective in RAMONES is to close the current marine radioactivity under-sampling gap. The RAMONES system concept, depicted in Figure 1, presents a new and efficient solution for in-situ, continuous, long-term monitoring of radioactivity in harsh underwater environments. Radioactivity measuring devices, so called gamma sniffers, will be mounted on underwater and surface vehicles, providing for the first time a mapping of the radioactivity distribution in the water column.

The Medusa stack, and the FAROL set of ROS nodes, are the base for the software running on the RAMONES system, abstracting the specificities of each vehicle, and providing a uniform way to access the system state, communication channels, and radioactivity sensors. The consortium has settled on equipping the robotic assets with Raspberry Pi 4 computers to run the RAMONES system software. These are small (but very capable) low-power single-board computers with multiple hardware interfaces that can be used to run the navigation and control algorithms, interface with the acoustic modems for communication and positioning, and obtain and processing data from the radioactivity sensors.

The nodes of the RAMONES concept are to be equipped with underwater acoustic (UWA) modems and modem-USBLs to provide communication channels among the different nodes and positioning of the AUGs either in relation to the ASV or to the benthic station, or to both simultaneously. This report focuses on the integration of the UWAs at the systems level, as the specifics of the positioning and communications were the subject of Deliverable 2.1.

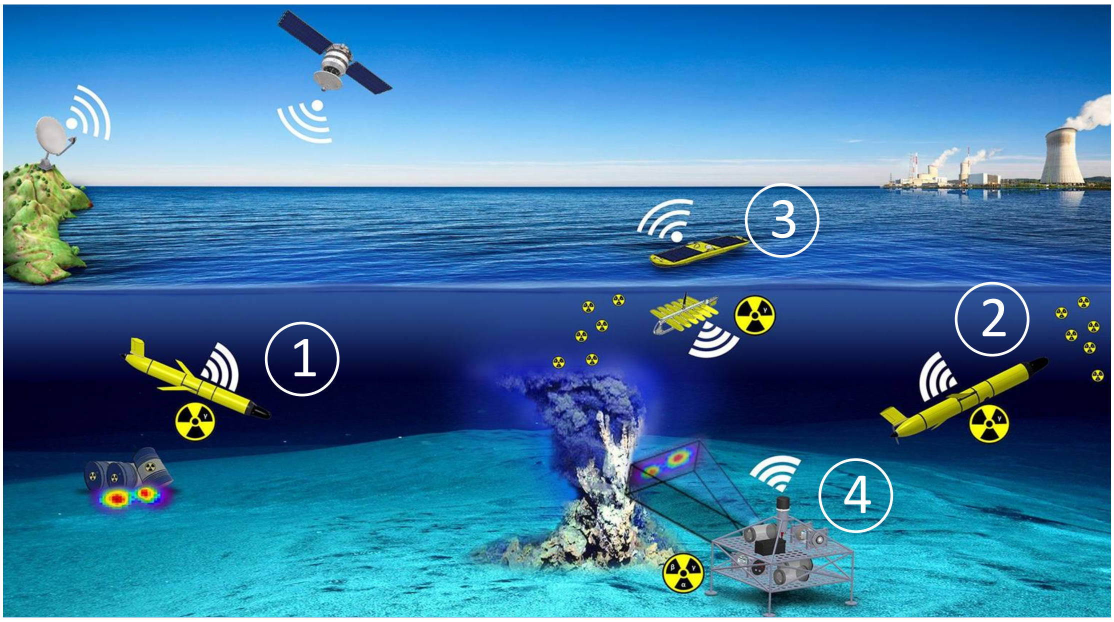

Figure 1: RAMONES concept showing the two AUGs (1,2), an ASV (3) and a benthic station (4).

2.1.1. Autonomous Underwater Gliders

The consortium decided on two Slocum G3 Gliders [4] for the role of mobile underwater platforms. These are autonomous vehicles with long endurance (they can be deployed up to one month) and proven performance and robustness. The differentiating feature of the Slocum G3 with respect to other commercial glider vehicles is the ability to programmatically interact with its onboard computer and modify mission parameters while the vehicle is deployed, through a so-called *backseat driver* (BSD) interface. This is a powerful and essential feature for RAMONES, as the network of vehicles is required to adapt in real-time to radioactivity measurements. Other gliders in the market can only be mission updated when the glider surfaces and obtains communications though a satellite link.

The Slocum G3 BSD interface is a specific protocol over serial port communications that allows for several glider sensors to be read (i.e., altitude, pressure, etc.) and mission parameters to be changed (i.e. waypoint, heading direction, etc.), up to 64 total variables. The Raspberry Pi 4 computer running the Medusa software stack acts as the *backseat driver* and communicates with the glider's science computer (ultimately connected to the flight computer) using the serial port. For ease of integration, RAMONES made use of the ROS package slocum_glider_extctl [5] which implements the BSD message protocol and makes the sensor measurements and mission parameters available as ROS messages. This allowed the Medusa stack to communicate and control the glider natively, i.e., within the ROS framework.

The gliders are to be additionally to be equipped with two hydroacoustic modem units, one of them including a USBL. The USBL unit is used to position the glider with respect to the surface vehicle, whereas the other is used for communication with the fixed benthic station. A block diagram of the physical connections between the RPi 4 computer, glider, and modems is presented in Figure 2.

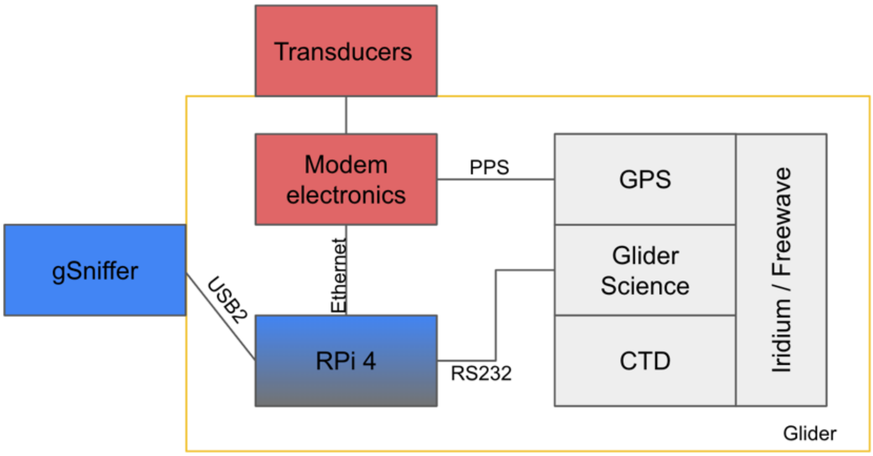

Figure 2: Block diagram of the glider internal physical connections between the gamma sniffer sensors, Raspberry Pi 4 computer, modem-USBL, and the glider science computer (BSD).

Report on Systems Integration and ValidationReport on Systems Integration and Validation

2.1.2. Autonomous Surface vehicle

For the ASV platform, RAMONES partnered up with ROC-SIANI, a division of University of Las Palmas de Gran Canaria's research institute SIANI, with which an MoU had been signed governing the use of the ROC-SIANI's A-Tirma as the ASV for the RAMONES project.

ROC-SIANI has extensive experience developing autonomous marine surface vehicles propelled by wind, so-called sail-drones or autonomous sailboats, and has currently two prototypes of these vehicles in operation, A-Tirma G2 [6] and A-Tirma G3.

In line with RAMONES's objectives, at the end of the project, the saildrone will be able to

- track trajectories that can be predetermined or issued on-the-fly based on commands from a shore station or onboard computing system;
- provide localization updates to underwater gliders;
- relay commands and data between gliders and a shore station;
- perform radioactivity measurements;
- perform multiday continuous operations.

The block diagram in Figure 3 shows the relevant internal blocks of the ASV hardware and the physical connections between the RAMONES ASV Box (containing a Raspberry Pi 4 computer and the modem-USBL's electronics) and the Saildrone internal navigation and communication systems. The saildrone is equipped with 4G and other radio connections for shore communication. Moreover, it has a GPS GNSS system with an available PPS output, required for the silent USBL system. It also provides a ROS interface to issue commands and receive localization and other telemetry data regarding the ASV. Finally, it is equipped with a gamma sniffer radioactivity sensor.

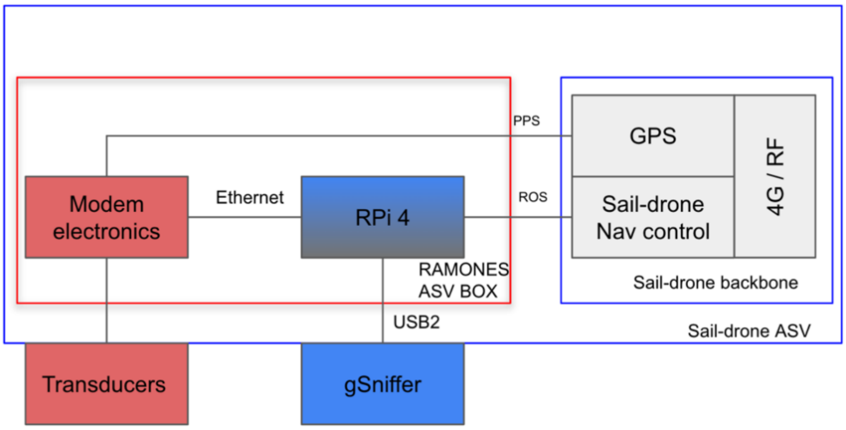

Figure 3: Block diagram of RPi 4 and ASV internal connections.

Given the geographical distribution of the partner teams involved, the development of the ASV is divided in 3 phases:

- Phase 1 - Integration and development testing at ROC-SIANI, ULPGC, Gran Canaria.
- Phase 2 - Development and operational testing, Lisbon.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

- Phase 3 - Operational demonstration, Santorini.

ROC-SIANI, ULPGC operates the A-Tirma G2 sail-drone, which is 2 meters long, and a G3 vehicle (3m) in development. G2 and G3 will share the same software so G2 will be used until G3 is ready for field operations. It is expected that the G3 will be used in the final demonstration to Santorini, albeit, if possible, it will be utilized as soon as possible in previous demonstration sites.

Figure 4: Payload bay of A-Tirma G3 ASV.

Figure 5: A-Tirma G3 ASV fully assembled.

Phase 1 is the ongoing integration and development testing at ROC-SIANI, ULPGC, Gran Canaria. Detailed integration specifications have been determined between RAMONES and ROC-SIANI, the vehicle is being robustified for longer endurance and upgraded to be able to accommodate the additional hardware, and a Hardware-in-the-loop simulator for the A-Tirma G2/3 has been made remotely available to the RAMONES consortium so that software integration can proceed before any field tests. The A-Tirma saildrone has two microcontroller units with the navigation subsystem separate from the communications subsystem. These communicate over CANBUS natively and external commands for control issuance and communications with peripherals can be issued, using A-Tirma's library can_lib, as illustrated in Figure 6. A ROS interface node, abstracting the CANBUS communications and can_lib library, will be provided by ROC-SIANI to facilitate the integration with the RAMONES system based on the Medusa stack. This interfacing node runs on the RAMONES Raspberry Pi 4 and will access the CANBUS through a Raspberry RS485-CAN bus hat.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

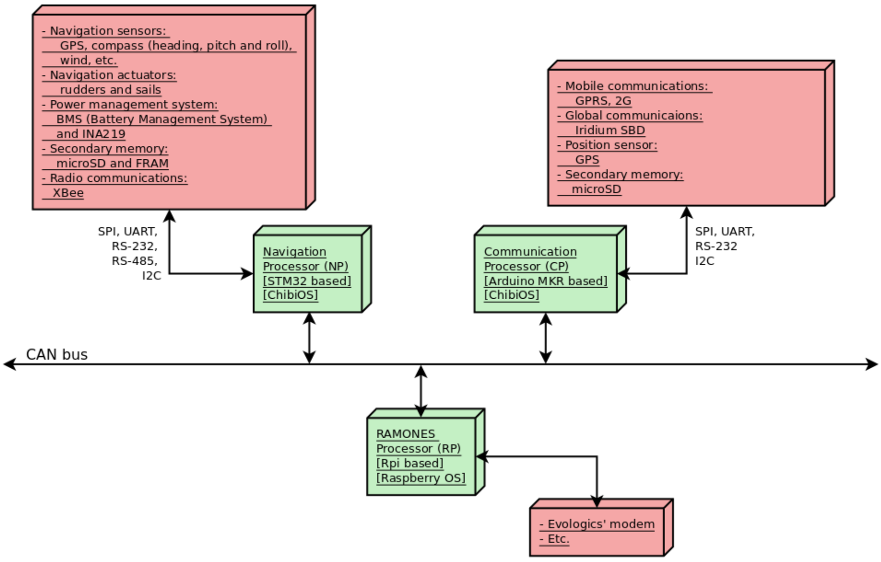

Figure 6: Block diagram of internal ASV architecture and its CAN bus link to the RAMONES Raspberry Pi computer.

Phase 2 includes the development and operational testing in Lisbon, in cooperation with IST-ID, where the saildrone will perform Cooperative Path Following (CPF) with a simulated glider at first, followed by CPF with an actual glider. This operation will demonstrate the Medusa stack communications and control architecture as well as the ASV integration with the USBL modems and other embedded systems.

Phase 3 will be the operational demonstration at Santorini area, Greece, where the whole RAMONES concept of cooperation between an ASV and two gliders is expected to be successfully demonstrated.

2.1.3. Benthic Station

The Benthic station is a static robotic asset that renders the navigation and control nodes redundant. The main job of the computational system onboard is to perform the measuring and logging of radioactivity sensors and process their data. Summary sensor data, as well as USBL positioning measurements from the mobile vehicles in range, will be relayed to the glider and surface vehicles over the acoustic channel network.

2.2. Medusa software stack and Evologics modem integration

Task 2.4 oversaw the development of the software interface between the Medusa and NetMARSys stacks and the Evologics' modems and modem simulator. This involved the creation of software ROS nodes to

- manage the processes of sending and receiving raw data from the acoustic modems;
- serialize and deserialize (convert to/from a byte stream) relevant ROS messages passed between the acoustic network assets;

Report on Systems Integration and ValidationReport on Systems Integration and Validation

- extract the acoustic message with the surface vehicle position, obtained from a Global Navigation Satellite System (GNSS);
- extract the USBL relative position measurements, combine them with the GNSS position of the ASV to obtain the global vehicle position measurement, and convert it to a format usable by the Medusa navigation filter;
- include the final USBL positioning measurement as an input to the navigation filter, where the new measurement will be used to refine the current estimate of the global vehicle position.

Figure 7 and Figure 8 illustrate the connections and message passing between the above-mentioned ROS nodes. The interface with the modem hardware and simulator is made through the Evologics' dmac ROS node. The diagrams refer to the integration of the Medusa stack and Evologics' modem software as used at IST-ID for preliminary experimental tests with existing vehicles and ping-reply interrogation scheme for the acoustic modems. Experimental results of coordinated maneuvers of multiple robotic vehicles using these connections and software are presented in Section 3. The final design establishing the integrated RAMONES system and the internals of the RAMONES marine robotic assets is detailed in Section 2.4.

The Evologics modem simulator allows for the creation of several virtual modems, each with a configurable location and properties of the acoustic channel, presenting themselves to the ROS system in the same way as an actual modem, through an IP address and sending and receiving ports for each acoustic channel.

To complete the fully integrated simulation system, the ROS node pos2simmodem was created to update the simulated position of acoustic modems with the simulated position of the vehicle it is mounted on. The position of the modems is thus updated in real-time, giving way to the test of the integrated RAMONES system with nonlinear acoustic channel disturbances occurring at long distances and large depths, before actual deployment.

2.3. Modular and extensible architecture towards mission readiness

The integrated RAMONES system was designed to leverage the benefits afforded by the ROS middleware and its abstractions. The Medusa framework builds on ROS for vehicle independence and its modular design allows a straightforward adaptation to any mobile marine vehicle without code modifications.

Regarding navigation, a sensor fusion node is configured to fuse all available sensor measurements and refine the vehicle estimates in an optimal fashion. As an example, the ASV can make use of high precision direct GNSS position measurements, whereas the AUGs must rely on noisy USBL measurements. The degree of trust in the sensor measurements, and on the vehicle model, is encoded in the design parameters of the Kalman filter that composes the sensor fusion node, which then outputs optimal estimates for position and other vehicle states. Since the mobile platforms will be used to explore a wide area of the ocean, it might occur that USBL positioning measurements are not always available to AUGs. In this situation, the AUGs must rely only on their system model to update the position estimate. If the condition persists, the positioning errors can eventually grow unbounded, a critical occurrence. Fortunately, the navigation algorithms available within the Medusa stack can make use of embedded range measurements to improve the accuracy of the AUGs' position estimates. These range measurements come for free with each transmitted message from an acoustic modem and can be included in an extended Kalman filter to improve the glider's position estimates. Experiments showing this localization method based on ranges have been performed by IST-ID and are described in Section 3.2 – Simultaneous localization and pursuit (SLAP).

The trajectory planning and cooperative motion control components are also vehicle independent as they operate at a kinematic level, providing outer-loop velocity references to the vehicle's dynamics inner-loop. It is only this last control loop, the so-called inner-loop, that is vehicle dependent and must be specified differently for each kind of vehicle and actuator disposition.

As remarked in Section 2.1.1, AUGs are typically sold by the manufacturer as highly autonomous vehicles and with no possibility of mission adaptation during an underwater flight nor interaction with its onboard sensors. To achieve the RAMONES goals, the consortium secured access to a proprietary interface, named Backseat Driver (BSD), allowing an external (but onboard the AUG) computer to access sensors and the internal state of the vehicle and, equally importantly, to modify mission parameters on-line, in real-time. Vital progress has already been made in this direction, with existing proof-of-concept code that reads relevant state variables such as AUG estimated location, depth, attitude, among others. RAMONES code can now influence, depending on the mission type, the desired depth or yo-yo movement, the desired x-y waypoint or the desired heading, or control directly the vehicle rudder for more accurate and flexible motion control.

For the AUGs motion control, RAMONES will make use of the already existing inner-loops implemented onboard of the vehicle, and make use of the BSD to provide the guidance references as desired setpoints for the relevant vehicle states. An identical approach will be followed for the ASV control, where RAMONES guidance references will be passed to the ASV internal motion controller, already implemented and tested by ROC-SIANI.

The use of identical Medusa stack setups for each vehicle brings implementation advantages for ancillary systems such as communications. The interface for receiving positioning, range measurements, or even arbitrary data is identical for all vehicles, owing to the ROS code abstractions and code reuse for the communication layers. Moreover, considering that both the Evologics' modems and simulator present themselves on the computer network in the same way, as a TPC/IP IP:port interface, it is trivial to alternate between realistic computer simulations and the actual acoustic hardware.

The RAMONES system was designed to be extensible to an arbitrary number of nodes, even though the final demonstration will include only four communicating nodes (1 ASV, 2 AUGs, and the Benthic station). The network topology for a mission is the specification of which nodes can communicate with which other nodes. In the RAMONES system, the network topology is defined in the comms.yml file and the acoustic message format is codified in acoustic_data.yml file, which appears highlighted in the diagrams of Figure 7 and Figure 8. These files are the only setup required for establishing the acoustic channels in the RAMONES implementation. Although the provided examples only show the position measurements from the modem-USBL as input to the sensor fusion node, the implemented sensor fusion filter can subscribe and use any type of measurement (be it of position, velocity, orientation, etc.) from the available onboard sensors to improve the robotic vehicle state estimate.

2.4. Integration of RAMONES assets with Medusa software architecture

The final integrated RAMONES system block diagrams for each of the RAMONES assets are detailed in Figure 10 though Figure 12.

The proposed glider integration in Figure 10 shows the two modems (one with USBL and the other without) with a silent interrogation scheme, both connected to a data_serializer node for message encoding and decoding. The gamma sniffer sensor also connects to the data_serializer so the AUGs can communicate summarized information to each other and to the surface vehicle. Received USBL positioning messages are combined with the ASV position (relayed as a message through the acoustic channel) to obtain an accurate global position for the AUG, which is fused with other available sensor measurements to obtain the best estimation of the vehicle’s position. The estimated position of the AUG by the ASV is also used for sensor fusion, and the resulting estimate of the AUG of its own position is sent to the ASV so that it can be forwarded to the mission control station. The control node has access to the current system state and desired maneuver for the glider, from which it computes the heading angle or desired waypoint and uses the backseat driver glider interface to modify them for the current glider mission.

In the ASV architecture design, depicted in Figure 11, it is possible to see identical dmac, silent, and data_serializer nodes as in the AUGs. These correspond to the paired modem-USBL that also exist in the gliders. The ASV can thus measure the relative position of the AUG, which it forwards back to the AUG via the acoustic link, and receives AUG’s own position estimate. Relevant mission data is forwarded to a mission control station via the Ponte node for remote monitoring. Periodic path planning for the ASV is performed based on the mission parameters and with feedback from both AUG’s positions and gamma sniffer data. The updated ASV path is sent to the Sailboat for tracking via the ROS node that provides the ROS-CAN bus translation interface.

Similarly, in the Benthic Station it is possible to identify again the dmac, silent, and data_serializer nodes that are in the modem-USBL pipeline. Here the inertial position of the benthic station (measured upon deployment) is combined with the relative position measured by the USBL unit (through range, elevation, and bearing measurements) to obtain an estimate of the AUG’s position. This estimate is then relayed back to the AUG for further refinement of its position estimate.

Table 1: Overview of the ROS nodes in the integrated RAMONES system.

dmac	A ROS Node from Evologics to parse messages from the device to ROS.
silent	ROS Node that is responsible for interacting with the dmac node by receiving and sending acoustic messages. Assuming that a UTC with atomic precision is received from the modem device, the trigger to send acoustic messages in each modem is based on this time according to a predefined interrogation scheme. This node also calculates the range measurement.
data_serializer	ROS Node that is responsible for serializing and deserializing when sending or receiving an acoustic message, respectively. The Serialization and Deserialization process acts according to what is specified in the acoustic_data.yaml file. This node publishes the ROS topics with the AUG's estimated position. The gamma sniffer data is packed at this node and sent via acoustics from each AUG.
usbl2pos	ROS Node computes the Inertial (NED) position of each AUG, ASV, and benthic based on the USBL angles and range.
ROS_Parser	ROS Parser that deals with Sail Boat interface. Allows to communicate between IST Farol Stack and ASV stack.
ahrs2farol	ROS Node that converts the topic message from dmac to be according to the one accepted by the filter:
Filter	ROS Node that fuses all the measurements from the USBL, acoustic positions, and depth cells to output a filtered state vector of the AUG.
IST-GLIDER API	A ROS Node responsible for managing the interface between IST Stack and Glider SCi PC:
ROS Driver	ROS Node that parses the gamma sniffer sensor measurements to ROS messages.
Data Processing	Node that computes summary information from the gamma sniffer measurements, suitable for transmission to the ASV and mission control station via the acoustic channel.
gnss2state	Converts the Lat Long ASV position sent via acoustic and publishes a ros topic with ASV UTM's position.
Ponte	IST Web console where it is possible to see the position of several vehicles (2- gliders and ASV).
Path Planning	ROS Node that computes the control references for each glider according to the gSniffer data received.
dmac2farol	ROS Node that rotates usbl angles according to Benthic's orientation. The modem-USBL onboard the Benthic Station doesn't have an embedded AHRS.

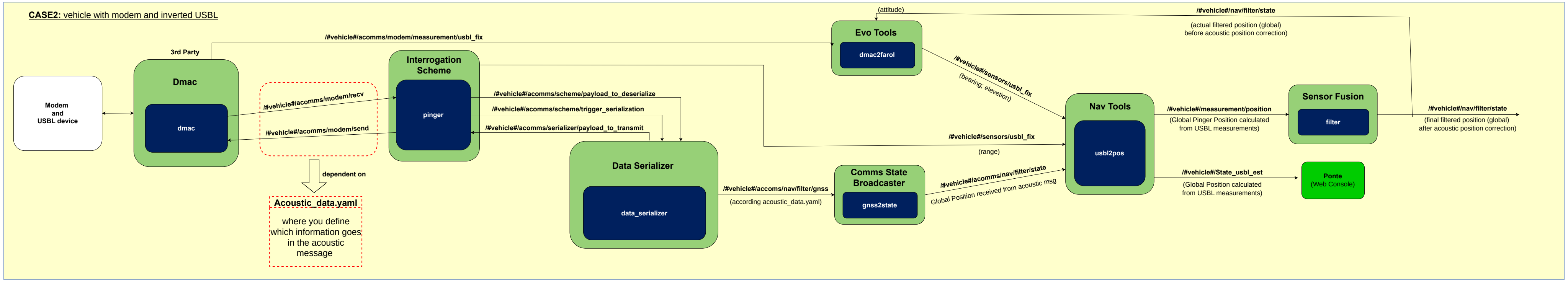

Figure 7: ROS message passing between Evologics modems and the Medusa stack.

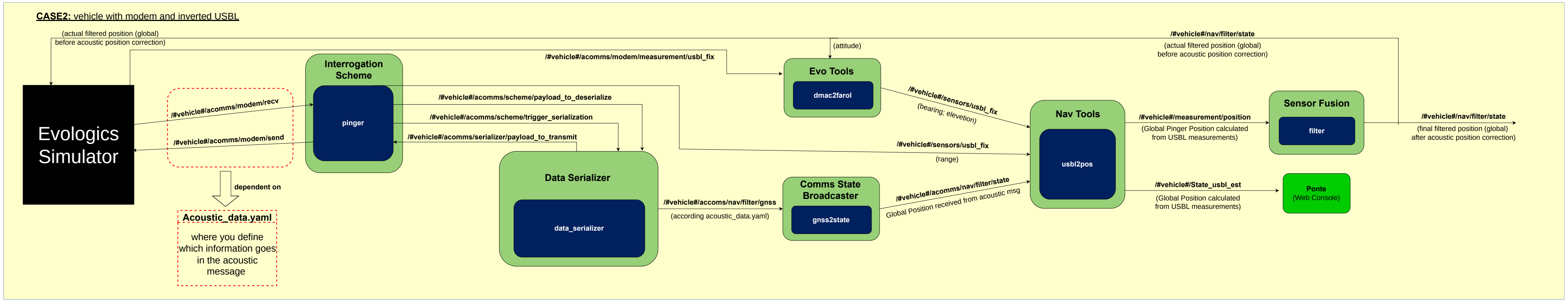

Figure 8: ROS message passing between Evologics modem simulator and the Medusa stack.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

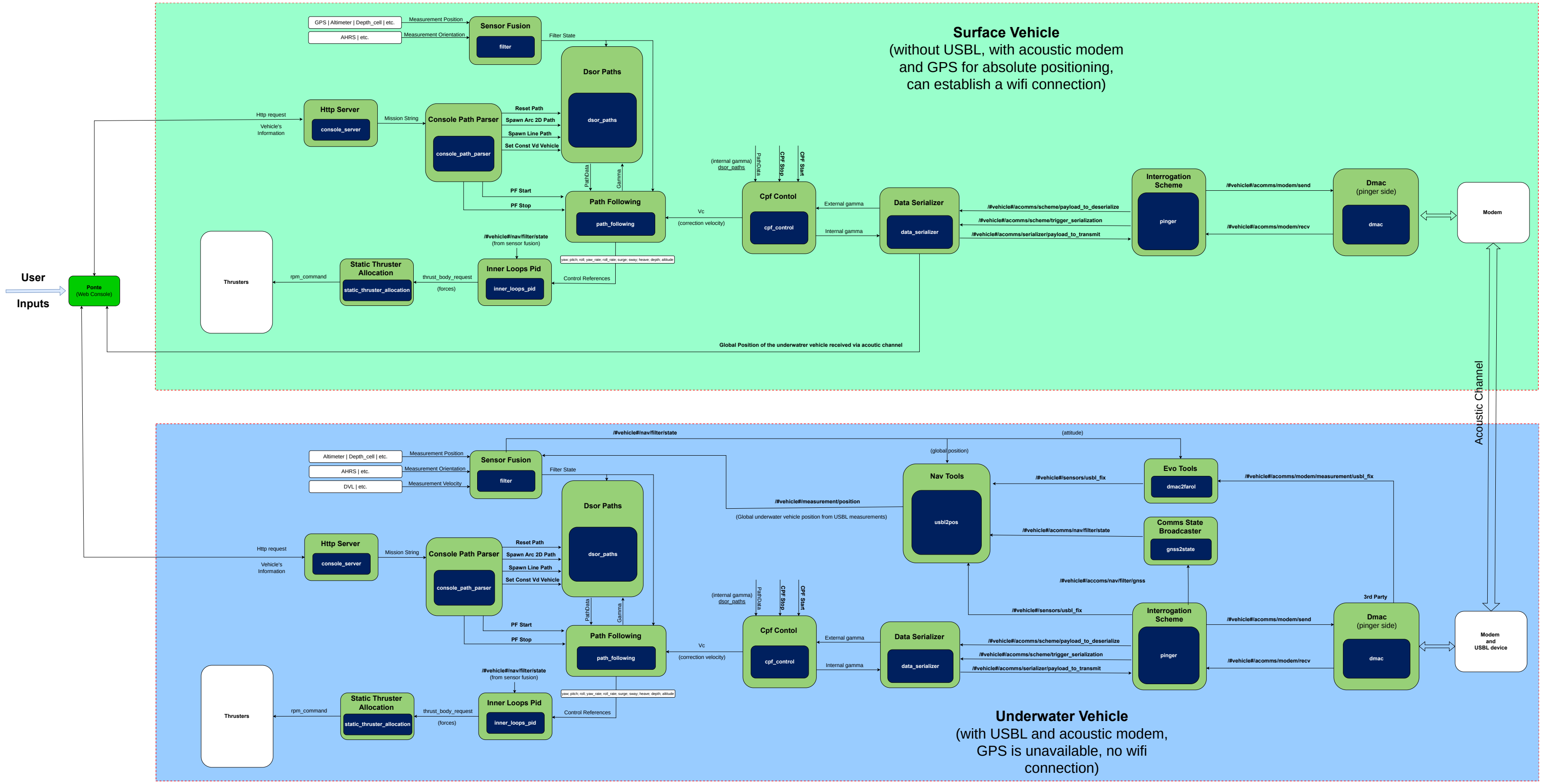

Figure 9: ROS message during Coordinated Path Following with acoustic modems.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

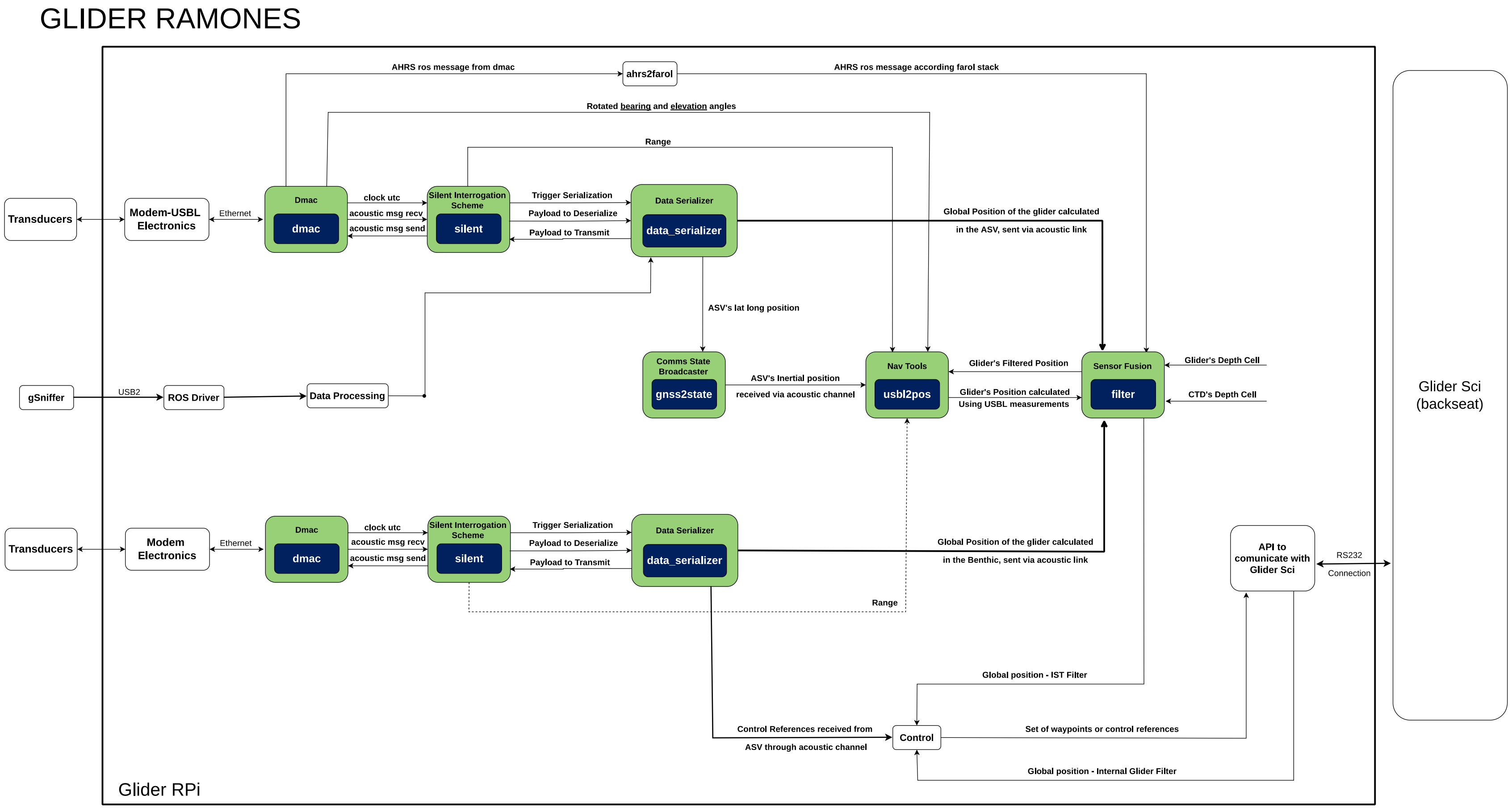

Figure 10: Block diagram evidencing the communication, navigation, and control ROS nodes for the AUGs.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

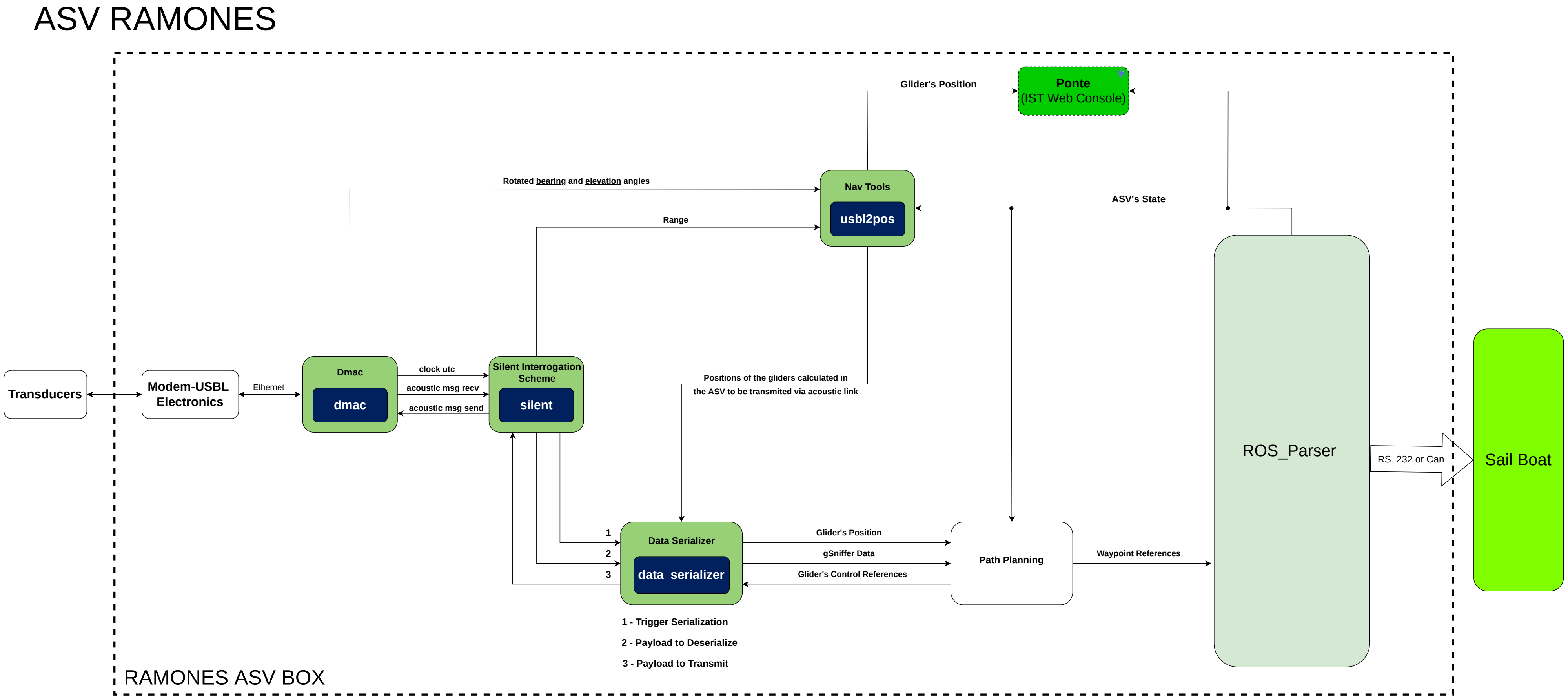

Figure 11: Block diagram evidencing the communication, navigation, and control ROS nodes for the saildrone ASV.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

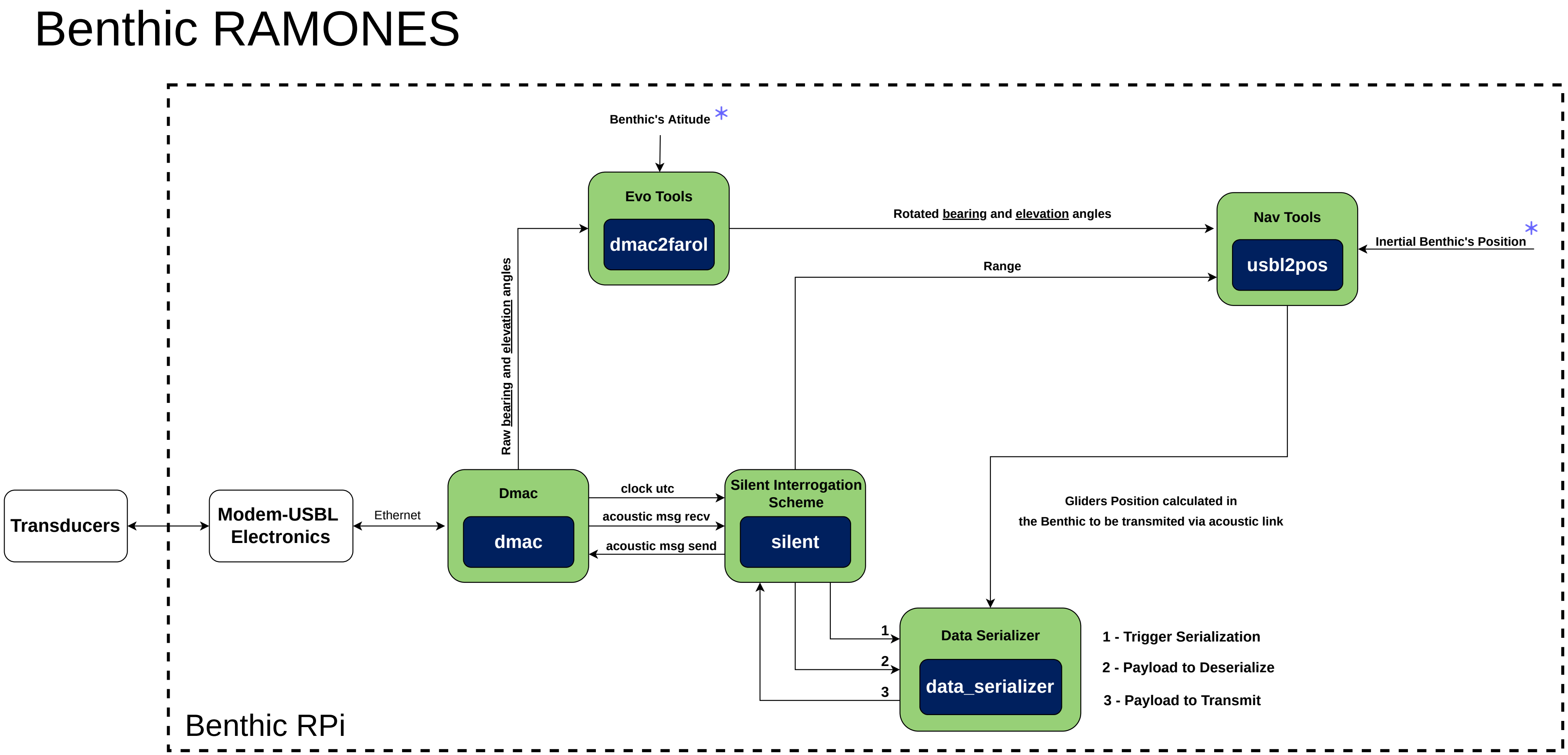

Figure 12: Block diagram evidencing the communication ROS nodes for the Benthic Station.

3. Experimental validation of integrated RAMONES system

As mentioned previously, acoustic communications are key components of the RAMONES project. A good pipeline must be established to let the agents of the acoustic network exchange data as needed. For a timely project pace, progress on WP2 tasks had to be undertaken before the glider and ASV vehicles were available to the consortium, as well as the Evologics modems and modem-USBLs with CSAC. As surrogates for the RAMONES vehicles and acoustic modems, the IST-ID Medusa vehicles were used, equipped with compatible Evologics modem-USBLs (but without CSACs). Two experimental tests, performed in the controlled environment at the Lisbon Expo dock, validating different aspects of the integrated RAMONES system, communication network, and positioning algorithms are described in the following subsections.

3.1. Missions with submerged vehicles and acoustics

To ensure that the acoustic pipeline is working properly, a series of Cooperative Path Following missions were deployed using the three MEDUSA vehicles available at the DSOR Lab. Two of the vehicles move at the surface level as ASVs (MEDUSA Black and MEDUSA Red) while the third vehicle (MEDUSA Vector, an AUV) is submerged, cooperating with the two surface vehicles. Cooperation is achieved by exchanging the path progression parameter between vehicles. The surface vehicles can communicate with each other using Wi-Fi, while communication with the underwater vehicle is available only through acoustics. The vehicles are all assigned a certain path to follow, and then start exchanging messages to achieve coordination among them.

Due to the harsh constraints imposed on the acoustic channel, the messages have a limited payload, so the information exchanged with each message is on a need-to-know basis. All things considered, the details shared between vehicles are the path progression to achieve coordination, the state of the underwater to the surface vehicles, monitoring flag bits to abort the mission if required, and the position of the surface vehicles to calculate the relative position of the submerged vehicle. The latter is absolutely necessary for the Ultra-Short Baseline sensor computations, turning the range, bearing and elevation measurements into a position for the submerged vehicle, which in turn make a correction to the only available sensor in underwater navigation, the Doppler Velocity Log (DVL) measuring velocities (dead reckoning). The vehicles are also all equipped with an acoustic transducer to send messages to each other.

The navigation filter of the MEDUSA Vector (mvector_nav_filter_state) can be seen in action in Figure 13 and Figure 14, predicting the movement of the vehicle and correcting the state of the vehicle with each USBL measurement (mvector_usbl_meas), along with the navigation filter of the surface vehicles (mblack_nav_filter_state and mred_nav_filter_state). These vehicles have much better localization than the underwater vehicle, since they are aided by GPS measurements. In Figure 14, the influence of the USBL measurements on the navigation filter of the submerged vehicle is perfectly visible, as well as the achievement of coordination between vehicles.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

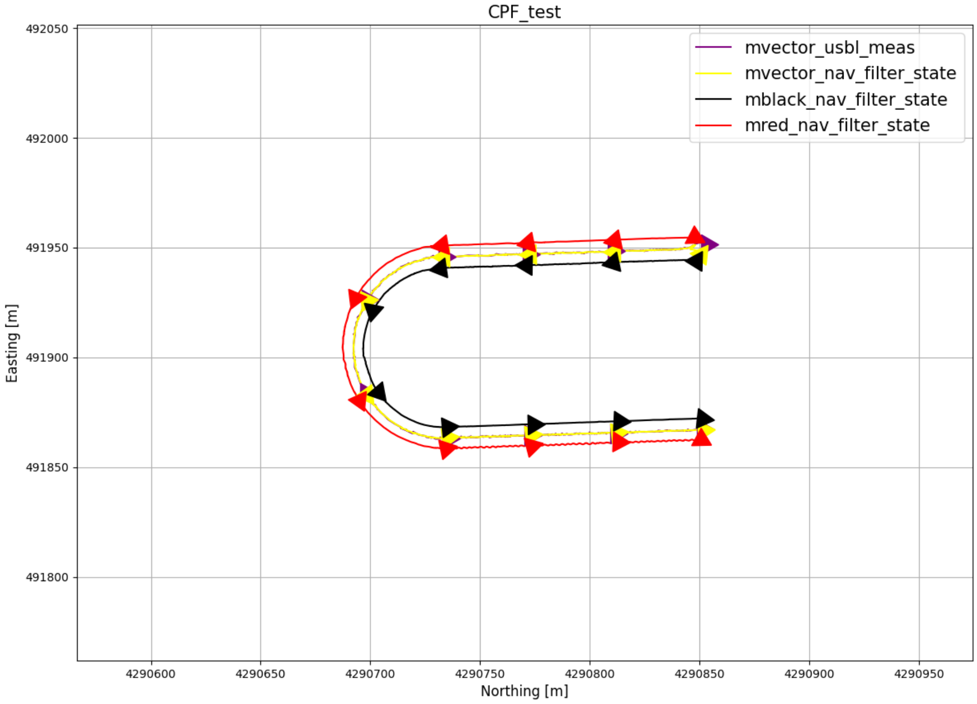

Figure 13: Cooperative Path Following with two surface vehicles (red, black) and one submerged vehicle (yellow).

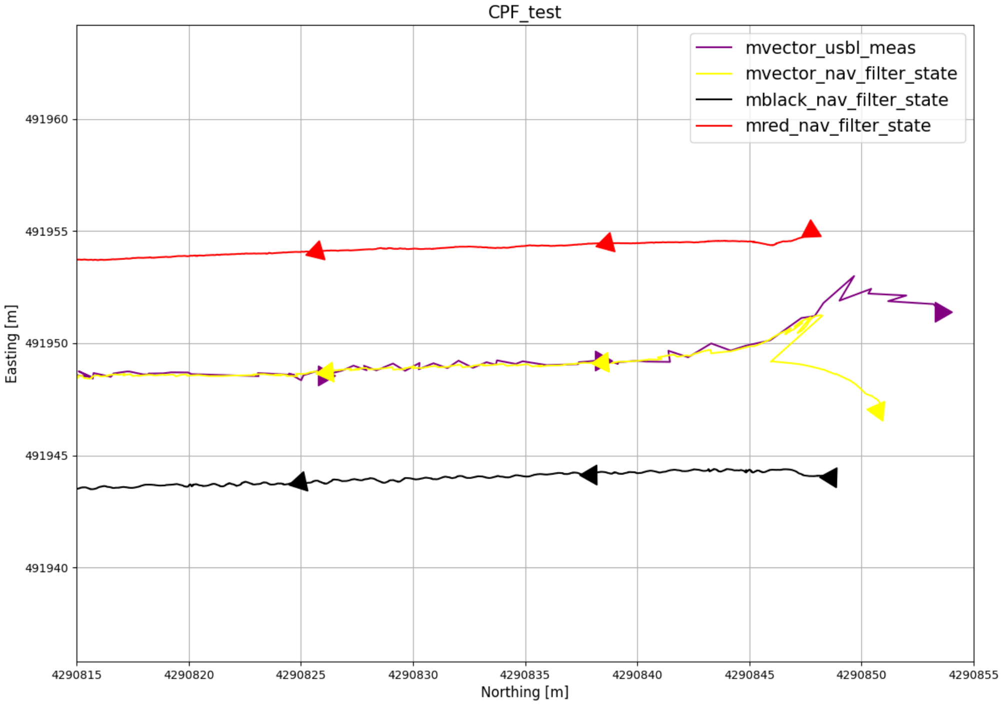

Figure 14: Detail of Cooperative Path Following with two surface vehicles (red, black) and one submerged vehicle (yellow). USBL positioning measurements in purple.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

3.2. Simultaneous localization and pursuit (SLAP)

The SLAP algorithm is an excellent way to see the previously described functionalities working in tandem. As the name suggests, two vehicles (trackers) work together to locate and pursue a given target. For these trials, the SLAP algorithm was fully integrated in the NetMARSys stack, simulating different scenarios to safely take the system into the real vehicles. In contrast to what had been done before, the target was accounted to be fully submerged, creating a great scenario to test the acoustic pipeline needed for the RAMONES project.

The position and velocity of the target is estimated by each of the tracker vehicles using an Extended Kalman Filter (EKF). Since a constant velocity model is assumed, the trackers predict where the target should be, given previous information, and then correct said prediction whenever a range measurement is taken from an acoustic modem. Finally, the estimates of each tracker are shared between vehicles to help the global estimation of the target, this type of filter being called a Distributed EKF (DEKF).

Now that the trackers know where the target is, an encircle maneuver is executed to pursue and estimate even better the state of the target, using a control law to converge the vehicles to a desired position while taking time constraints into account. This way, the trackers pursue the target in a ninety-degree formation to achieve the best estimates for the DEKF. This specific formation was determined to be the best for two tracker vehicles trying to pin-point a target. A behind target formation is also implemented, as it is observed in the Figure 15, where the vehicles maintain a ninety degree angle while pursuing the target from behind.

The next step to achieve the correct formation between trackers is to cooperate and exchange information about their path progression. Once coordination is achieved, both trackers should maintain the ninety-degree formation with ease. Since in the trials the trackers are surface vehicles, they can easily share the information via Wi-Fi. Event-Triggered Communications (ETC) are also implemented to reduce the number of messages to a minimum, only triggering when there is a significant error between both vehicles.

The real scenario was executed using the MEDUSA Black and MEDUSA Red as surface trackers, and the MEDUSA Vector as a submerged target performing a simple lawn-mower Path Following maneuver, completely unknown to the trackers. The underwater vehicle is monitored closely via acoustic communications with the surface vehicles, which in turn ping continuously the target to help its self-localization. This way, range measurements between trackers and target are constantly taken in parallel, giving way to the estimation of the position of the underwater target.

In Figure 15, the path coursed by MEDUSA Red, MEDUSA BLACK and MEDUSA Vector are shown, as well as the USBL measurements by the MEDUSA Vector using both trackers as anchors to compute its position. Furthermore, the estimation of the target made by each tracker is represented with an uncertainty cloud plotted around the target. The mission is divided in three phases, clearly seen in Figure 15 (left): first the vehicles encircle the target as it starts converging to the assigned lawn-mower path; after completing the first section of the path, the vehicles change to a behind target formation, keeping up the localization and pursuit; finally, after the first turn is completed, the underwater target increases its surge speed, making the trackers step up their speed in consequence to keep up with the target. The underwater localization with the aid of the USBL sensor is again proven to work well with

Report on Systems Integration and ValidationReport on Systems Integration and Validation

the navigation filter of the submerged target, as well as the estimates by both surface vehicles using only acoustic range-measurements of said target, as seen by the uncertainty clouds around it as the mission unfolds. The performance in open-water is expected to be much better but, in any case, the DEKF filter was able to keep a good estimate of the position of the target.

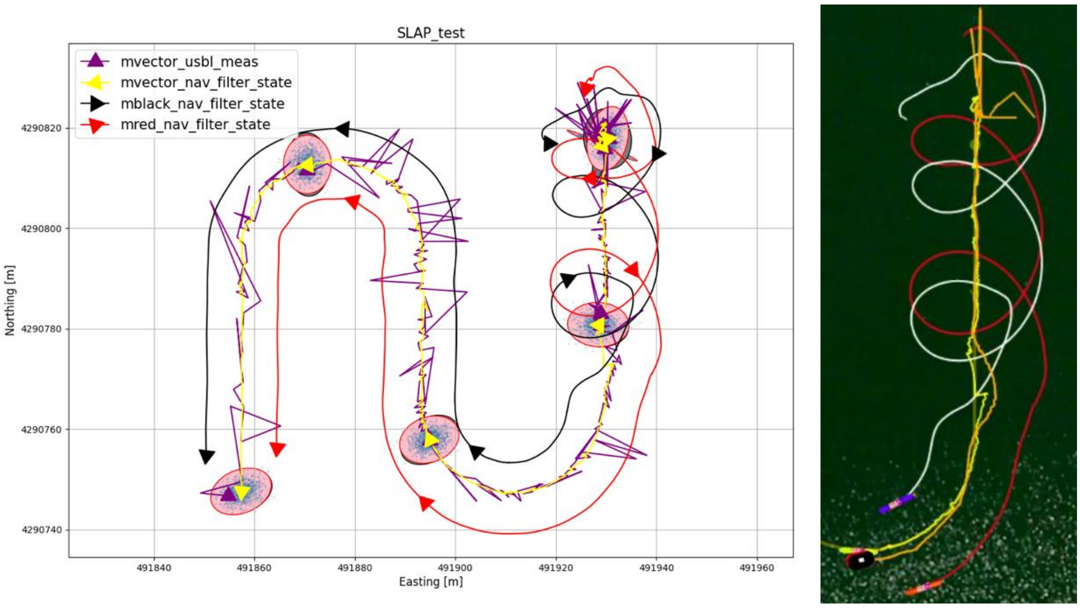

Figure 15: Overview of a SLAP maneuver with overlayed positioning uncertainty for the target (left). Detail of a command console screen capture showing the three vehicles and the uncertainty cloud for the target (right).

Report on Systems Integration and ValidationReport on Systems Integration and Validation

4. Initial experimental tests with the glider vehicles

Iteratively perform many engineering trials during the development phase of a project like RAMONES is of the utmost importance for assessment and refinement of the project design decisions. Doing the tests in the water, in a controlled environment, with the Slocum gliders – to be equipped with radioactivity sensors – is particularly difficult. This vehicle was designed and optimized for deep diving up to 1000 m, requiring a minimum water column of 20 to 30 meters to operate in the normal repetitive yo-yo mode (dive-surface-dive-(...)) using variable buoyancy as its *engine* to propel it forward. Furthermore, it would be extremely convenient to have GNSS position measurements during the initial trials (for ground truth) and have a continuous radio link established with the vehicle for monitoring its internal state and debugging purposes.

For this reason, IST-ID designed and performed several modifications to the glider vehicle such that it remains all the time at a very well-defined depth, while maneuvering, with the objective that the GPS antenna remains just out of the water and the rudder remains as submerged as possible. The modifications can be seen in Figure 16 and consist of:

- A contraption providing additional flotation and additional ballast.
- A rudder extender, to be attached to the rudder, in order to increase the vehicle maneuverability (reduce the minimum turning radius).

Figure 16: Glider equipped with contraption for added floatability (left) and detail of rudder extender (right).

These modifications were implemented and first tested during the trials in a test tank on October 21st, 2022, see Figure 17. Everything went as expected. The second tests took place on the October 26th, see Figure 18, deploying the glider for the first time in salt water,

Figure 17: Initial tests in a tank.

Figure 18: AUG sailing at the expo site.

Report on Systems Integration and ValidationReport on Systems Integration and Validation at the EXPO Dock, in Lisbon. The goals were to test the radio link, the GPS and the maneuverability of the vehicle while it remains just under the surface.

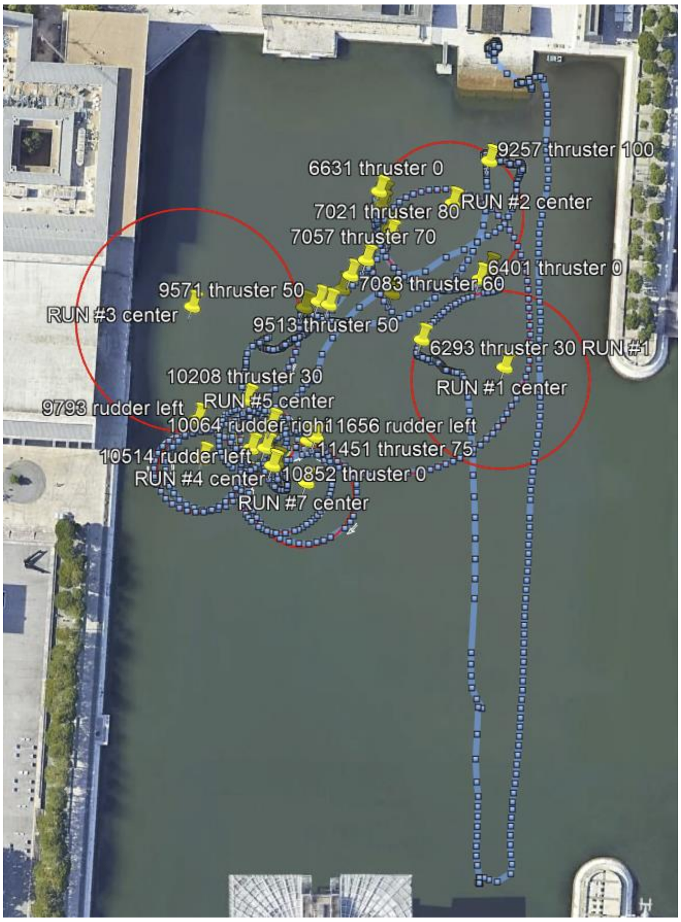

Figure 19: GPS data points during the trials. The range test (top-to-bottom segment and back) and several arc-circles (full circles overlayed in red) are clearly visible.

The results, for which GPS data points are presented in Figure 19, can be condensed as follows:

- The radio link worked all the time for the whole extent of the dock, up to 345 m from the shore station.
- The GPS worked all the time. The GPS antenna was out of the water 100% of the time, as long as the water remained flat (if a boat passed by, then the GPS antenna was submerged for brief moments).
- The vehicle could achieve a speed of 0.70 m/s using a thruster command of 75%.
- The vehicle shows a non-negligible tendency to turn to the right with the rudder at the center, doing a turning radius of about 35 m. As a consequence of this, the results obtained when the rudder is deflected to the left or to the right differ a lot.
- With the rudder extender installed, the average turning radius is around 15 m.
- Without the rudder extender installed, the average turning radius is more than 30 m.

Report on Systems Integration and ValidationReport on Systems Integration and Validation

The main conclusion is that the proposed solution enables the RAMONES partners to perform trials with the gliders at the surface as long as the surface of the water is really flat (like a dock or other controlled environment without waves).

4.1. Validation of setup for surface navigation

The contraption providing additional flotation and additional ballast proved a success, with the following being observed during the trials:

- The **radio link was OK** during all trials at the surface for the whole area of the dock.
- The **GPS is able to get fixes** during all trials at the surface.
- The speed characteristic as a function of the thrust command is approximately linear, as expected: 30% thrust => 0.33m/s; 50% thrust => 0.50m/s; 75% thrust => 0.70m/s;
- During the curves (rudder deflected at max left or max right) the speed reduces slightly (3% to 6% reduction) when compared to the condition with the rudder at the center.
- Several gliders turning maneuvers were performed at different speeds resulting in the average turning radius detailed in Table 2.

Table 2: Turning radius for different glider configurations.

Rudder extender	*Propeller thrust*	*Average Radius [m]*
with	30% Thrust	16
with	75% Thrust	15
without	30% Thrust	40
without	75% Thrust	36

The preliminary conclusions about the rudder extender are:

- it **significantly improves the turning performance of the glider**: if the expected performance shown above is correct, then the radius will be reduced by 59% when the rudder extender is added;
- it **is mandatory in order to be able to run experiments in a confined space** such as the Expo dock.

During the trial tests, it was verified that, with the additional contraption, the vehicle has a **tendency to turn right which cannot be neglected**. This affected the results and would possibly affect any attempt to run autonomous missions with the glider because the performance would depend a lot on the direction of turning. In order to minimize or eliminate that effect **it was decided to design and install adjustable trim tabs**, such that the vehicle goes straight when the rudder is at the center. For space and symmetry reasons two trim tabs are used, fixed on the posterior section of the additional floatation contraption. These are installed at a fixed angle of -5 degrees and have a mobile component that can be adjusted +/-

10 degrees for fine-tuning, which is fixed in place with a standard screw. An illustration of the trim tab mechanism is provided in Figure 20.

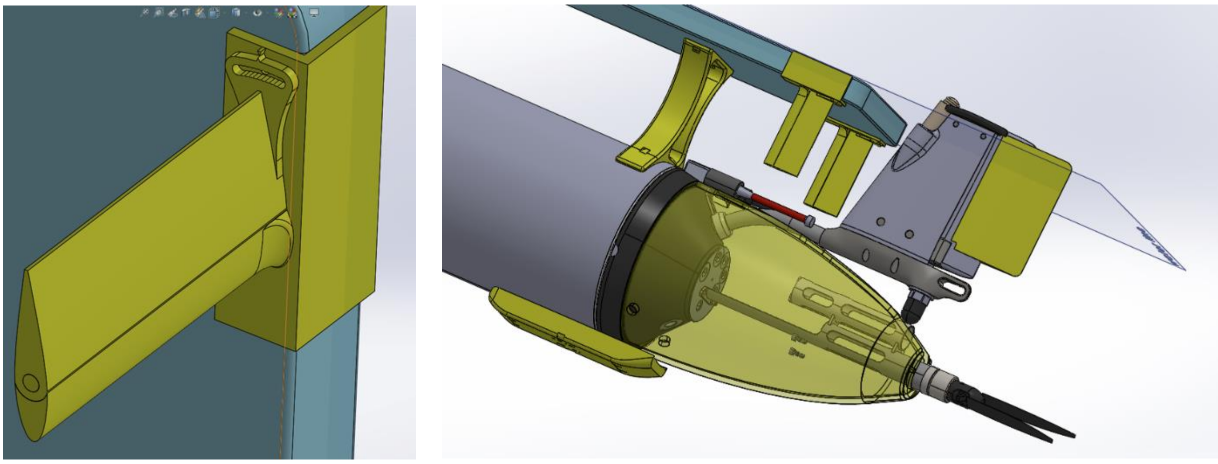

Figure 20: Close up of the trim tab (left) and its location on the floating contraption (right).

The added flotation device as well as the trim tabs are poised to enable the necessary extensive testing of adaptive sampling and navigation algorithms for the AUGs in a safe and confined environment where there is no risk of outright losing the vehicle. Having the integrated RAMONES system tested and verified to be working as designed, not only in HITL simulation but also with the real vehicles in a controlled environment, is fundamental for a first successful mission at open ocean and at a large depth such as the envisioned Kolumbo volcano.

References

[1] P. C. Abreu *et al.*, "The MEDUSA class of autonomous marine vehicles and their role in EU projects," *OCEANS 2016 - Shanghai*, Shanghai, China, 2016, pp. 1-10, doi: 10.1109/OCEANSAP.2016.7485620.

[2] S. Garg, J. Quintas, J. Cruz and A. M. Pascoal, "NetMarSyS - A Tool for the Simulation and Visualization of Distributed Autonomous Marine Robotic Systems," 2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV), St. Johns, NL, Canada, 2020, pp. 1-5, doi: 10.1109/AUV50043.2020.9267922.

[3] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat and T. Rauschenbach, "UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation," OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 2016, pp. 1-8, doi: 10.1109/OCEANS.2016.7761080.

[4] Teledyne marine webpage, http://www.teledynemarine.com/slocum-glider

[5] Gitlab repository for the slocum_glider ROS metapackage, https://gitlab.com/sentinel-aug/ros/slocum_glider

[6] Domínguez-Brito, A.C., Valle-Fernández, B., Cabrera-Gámez, J., Ramos-de-Miguel, A., García, J.C. (2016). A-TIRMA G2: An Oceanic Autonomous Sailboat. In: Friebe, A., Haug, F. (eds) Robotic Sailing 2015. WRSC/IRSC 2015. Springer, Cham. https://doi.org/10.1007/978-3-319-23335-2_1

Report on Systems Integration and ValidationReport on Systems Integration and Validation

List of acronyms

Acronym	Description
HITL	Hardware-In-The-Loop
AUG	Autonomous Underwater Glider
ASV	Autonomous Surface Vehicle
ROS	Robot Operating System
API	Application Programming Interface
CSAC	Chip-Scale Atomic Clock
USBL	UltraShort BaseLine
SITL	Software-In-The-Loop
FAROL	Free Autonomous Robots for Observations and Labelling
PONTE	Page for Operation, Navigation and Tinker Engagement
UWA	Underwater Acoustic
SiNAPS	S2C Intelligent Navigation And Positioning Software
BSD	BackSeat Driver
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
CAN	Controller Area Network
CPF	Cooperative Path Following
SLAP	Simultaneous Localization And Pursuit
EKF	Extended Kalman Filter
DVL	Doppler Velocity Log
DEKF	Distributed Extended Kalman Filter
FIM	Fisher Information Matrix

Report on Systems Integration and ValidationReport on Systems Integration and Validation

Appendix 1