The MEDUSA class of Autonomous Marine Vehicles and their Role in EU Projects

Pedro Caldeira Abreu*, João Botelho*, Pedro Góis*, António Pascoal*,
Jorge Ribeiro*, Miguel Ribeiro*, Manuel Rufino*, Luís Sebastião* and Henrique Silva*
*Laboratory of Robotics and Systems in Engineering and Science (LARSyS),
ISR/IST, University of Lisbon, Lisbon, Portugal.
Email: antonio@tecnico.ulisboa.pt

***Abstract*—This paper describes a new class of autonomous marine vehicles named MEDUSA and highlights their role in a number of EU projects addressing multiple vehicle scenarios. The MEDUSA vehicles, with surface and diving versions, were designed and built at the Institute for Systems and Robotics, IST, Univ. de Lisboa, Portugal as a result of an effort aimed at affording researchers and practitioners of marine robotics tools to: i) assess the efficacy of cooperative motion control and navigation algorithms and ii) seamlessly execute scientific and commercial missions with multiple robots at sea.**

We first define the problem of designing the MEDUSA-class of vehicles by describing the desired functional requirements that motivated their development and then present our solution. Mechanical and electrical design considerations that relate to the requirements are explained, and the software architecture is described. This includes a brief overview of the navigation system, the main lower-level control loops which other features can build upon, and some of the higher-level algorithms for multiple-vehicle cooperative missions. We also illustrate the functionality of the mission control system, a dedicated software suite that allows operators to seamlessly program and follow the state of execution of cooperative missions involving multiple vehicles, possibly running different operations or missions in parallel. Finally, we summarize the participation of IST and the MEDUSA vehicles in a number of representative EC-funded Marine Robotics-related projects.

I. INTRODUCTION

The use of Autonomous Marine Vehicles (AMVs) has become commonplace for a wide range of civil, military and especially scientific activities. Numerous AMVs have been built by companies and universities in recent years, targeted towards a variety of different applications, which then translated into very different vehicle characteristics. This paper focuses on the MEDUSA class of AMVs, designed by the Institute for Systems and Robotics (ISR) of the Instituto Superior Técnico (IST) at Universidade of Lisboa, Portugal. The main intended purpose of these vehicles is to serve as an easily-available research tool for testing and exploring new algorithms in the context of autonomous control and navigation in marine environments, especially cooperative techniques that make use of several of these vehicles. The mechanical and electrical design process was heavily motivated and constrained by a set of envisioned operational scenarios for the vehicles, which translated into requirements and desired characteristics. Following the production of the first vehicle, the entire software architecture was written, including some primitive control and navigation features which higher level algorithms can build upon. In this paper we summarize these stages of the design and development process, leading up to the mature stage of development that the MEDUSAs have reached. We also highlight the roles that these vehicles and IST have played in a number of projects in the last years.

The main contributions of this paper are: i) an overview of the development process of the MEDUSA vehicles, and ii) a description of the instrumental roles these vehicles have played in the marine robotics research ecosystem through the participation in European Commission (EC) projects.

The remainder of this paper is organized as follows. Section II defines the problem to be solved in terms of desired or required characteristics of the MEDUSAs. In section III we look at the proposed solution for this problem, considering mechanical and electrical design, software development including control and navigation systems, and the console for mission control and visualization. Section IV focuses on the participation of the MEDUSAs and IST in EC-funded projects, including brief descriptions of the goals of the most representative projects and the results achieved. Section V provides some final conclusions and looks at possible future advances regarding the MEDUSA vehicles.

II. PROBLEM STATEMENT

The development of the MEDUSA class of vehicles was heavily influenced by its goal to afford marine robotics researchers a tool to easily prototype, test and explore algorithms for automatic control and navigation - possibly to be later implemented in more robust platforms, such as ROVs or industrial-grade AUVs. The extensions of these algorithms to multiple-vehicle scenarios was also envisioned. In these scenarios, a number of vehicles can exchange information and cooperate to improve the overall performance of the entire fleet in a variety of tasks. This section defines in clear terms the main high-level requirements that these goals imposed on the vehicle design.

A. Reduced cost

One of the main concerns when designing the MEDUSAs was to keep costs to a minimum. This was attempted mainly in two domains.

978-1-4673-9724-7/16/$31.00

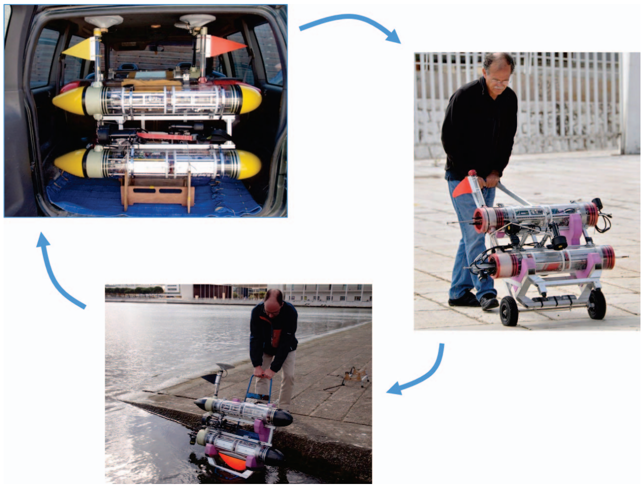


Fig. 1. Easy transportation of 3 Medusas in a small van or trailer, and launch and recovery using a cart.

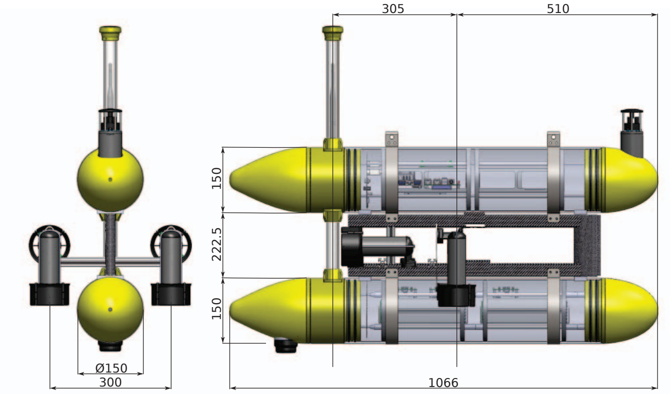


Fig. 2. A schematic illustration of the Medusa side and back views with dimensions in millimeters.

1) The components (especially the mechanical parts) were chosen with a heavy emphasis on keeping costs reasonable, to make the vehicles affordable for universities or research groups in general wishing to acquire them to use as research tools.
2) The logistics requirements and costs of conducting experiments were reduced to a minimum. This is crucial given the expected exploratory nature of most of the tests carried out by the vehicles, as research activities often imply. Less operational requirements allow for more frequent testing, hence facilitating the rapid development of new features. In practical terms, the vehicles should be small and light enough for two people to easily transport in a normal car, launch, recover, and perform the necessary post-operation maintenance, as shown in fig. 1.

Attempts to reduce costs on these two fronts imply roughly the same drawbacks: limitations in terms of autonomy (smaller vehicles imply less battery volume onboard), maximum depth (components that can stand bigger pressures are, in general, heavier and more expensive), and maximum payload.

B. Flexible payload changes

Research requires continuous adaptation and exploration of different approaches. The design should take this into consideration by i) reserving considerable room for complementary equipment, ii) allowing the onboard computer to interface with a large number of sensors, and iii) making this payload easily changeable.

C. Easy inter-vehicle communications

As stated previously, multiple-vehicle cooperation is one of the main research areas that the Medusas aim to support. As such, a key requirement is that the vehicle design takes into consideration the frequent need for communication, both via radio (e.g. Wi-Fi) and underwater acoustic channels. Specifically, radio communications should be available whenever the vehicle is operating at the surface, while acoustic communications should be available both when at the surface and when underwater. When operating at small depths, the use of a buoy carrying a Wi-Fi antenna connected to the vehicle with an underwater Ethernet cable is also possible, allowing communications with the on-shore mission control station at a very high rate. This is especially useful when developing and debugging new features.

D. Autonomy for 12 hours of operation

Autonomy is closely related to the size of the vehicle, as batteries occupy considerable volume. The research activities which we address usually do not require several days of continuous operation. As such, the autonomy of the Medusas at a typical load is set to about 12 hours. This is enough to avoid interrupting operations during a test day to charge batteries, and still allowed us to keep the vehicles small and light by avoiding excessive volume and weight of batteries.

E. Extensible software architecture

The software controlling the vehicles should offer enough abstraction for researchers to be able to easily implement new features without deep knowledge of the entire architecture. Low-level operations (drivers for interfacing with sensors and actuators, battery management system, and safety features) are transparent to users. Additionally, a number of basic control and navigation features (inner-loops to track speed, heading, and depth references, and a navigation filter to compute position and velocity estimates) should be provided for higher-level modules to build upon.

III. PROPOSED SOLUTION

A. The Medusa-class vehicles

At the moment, two different segments of vehicles have been developed: the Medusa$_D$ (with diving capabilities down to 50 m depth) and Medusa$_S$ (able to operate only at the surface). A schematic representation of a Medusa$_S$ vehicle is shown in fig. 2. The vehicle consists mainly of two cylindrical bodies made of acrylic, making the vehicle much lighter and cheaper at the expense of limiting the maximum depth of operation, as stated in section II-A. The two cylinders are connected so that cables can be passed between them, and held together by an aluminium frame. If desired, changing the distance between these two cylinders requires the production of only a few new parts of the frame. A larger distance increases the metacentric height and improves the performance of the acoustic sensors when at the surface, but impacts the logistics necessary for transportation and slightly degrades the hydrodynamic characteristics (i.e. increases drag).

The thrusters are placed on both sides of the vehicle. Two thrusters aligned longitudinally with the cylinders provide forward thrust force and/or yaw moment, and no rudders are used. For the MEDUSAs with diving capabilities, two additional thrusters aligned vertically provide vertical thrust for diving and surfacing.

Equipment carried by the vehicles can be categorized as core or optional payload. Figure 3 illustrates both the core equipment onboard the MEDUSAs and all the optional payload sensors that have been tested since the first vehicle was built.

B. Mechanical design considerations

The most striking characteristic of the MEDUSAs is the two-cylinder structure. This kind of two body shape serves three main purposes, which relate in part to the requirements stated in sections II-B and II-C:

1) increasing the metacentric height by placing the batteries in the lower cylinder, thereby lowering the center of mass and improving stability in roll and pitch motions;
2) decoupling “air and water segments” by placing the radio and GPS communications equipment on the upper tube and all the wet sensing systems (including acoustic communications) in the lower tube, so that both dry and wet sensors work while operating at the surface;
3) facilitating the change of complementary payload depending on the mission at hand by reserving volume for generic sensors in the lower cylinder. They can be removed or added without the need to move the core components, which are mainly in the upper cylinder (except for the batteries). The two ends of the lower cylinder are particularly adequate to place wet sensors.

C. Electrical design considerations

The power source for the thrusters, computers, and all sensors is a set of two 7-cell packs of lithium polymer (LiPo) batteries, placed in the lower cylinder, working at 26V with a capacity of 32 Ah. These batteries have higher energy density and are safer than lithium ion, have longer life cycles and faster charging than silver zinc, and require little maintenance. On the other hand, they require a strict charging regime with a CC/CV (constant current-constant voltage) cycle and a protection circuit to prevent over-discharge. The charging, equalization, and discharging processes are monitored by a microcontroller system *BatMonit*, developed in-house at IST.

The amount of energy carried onboard the vehicles is mainly a trade-off between autonomy (in terms of hours of operation) and the vehicle size and weight. The total battery capacity was chosen according to the requirement in section II-D, also taking into account the need for a light and small vehicle as stated in section II-A. Table I illustrates the autonomy of the vehicle at different speeds, with no considerable complementary payload.

TABLE I
CURRENT, VELOCITY AND AUTONOMY FOR DIFFERENT THRUSTER RPMS.

RPMs (%)	Current (A)	Velocity (kn)	Autonomy (h)
10	1.3	0.4	24
20	1.5	0.75	21
30	1.9	1	16
40	3	1.5	10
60	5.4	2.9	6

D. Software architecture

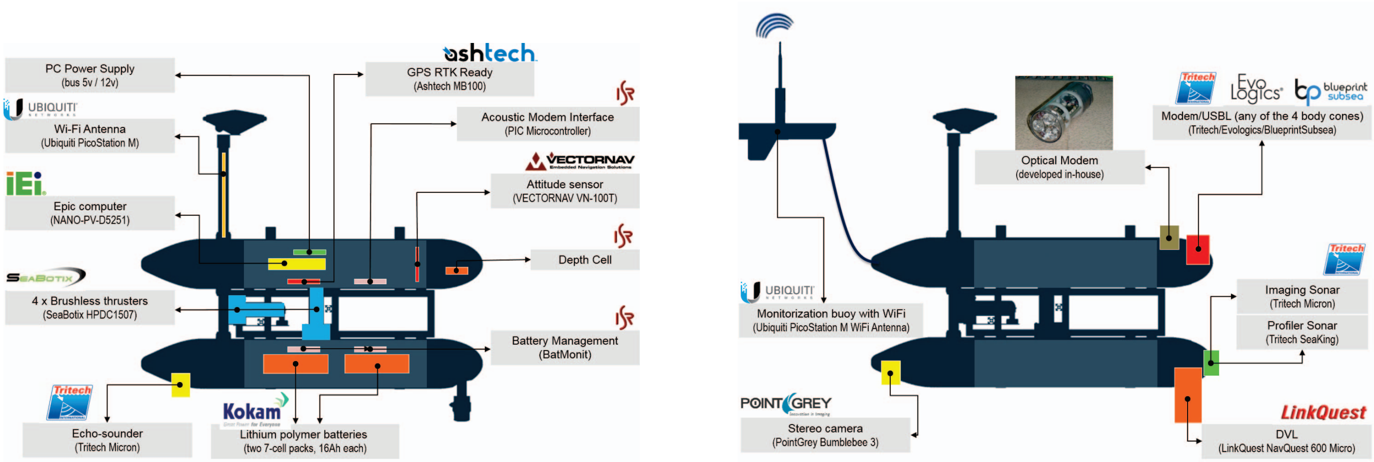
An overview of the software architecture running on the MEDUSA vehicles is represented schematically in fig. 4. It comprises five main blocks:

- the navigation filter, as described later in section III-E;
- a large control group, including both the inner-loop controllers and higher-level, possibly cooperative algorithms, as described later in section III-F;
- low-level software: drivers for the thrusters and all the sensors, and the battery management system *BatMonit*;
- a mission control and planning block which orders controllers to start or stop based on the mission specifications; at the moment there is no unified structure for this block, as it depends greatly on the project or application at hand;
- a communications layer that handles all transmissions from and to the vehicle, namely acoustic or WiFi transmissions to other vehicles and WiFi transmissions to and from the mission control console.

Most of the modules in this architecture are implemented using the Robot Operating System (ROS). It is based on the concept of *nodes* (small software modules), and *topics* (communication channels between the nodes), making the architecture easy to extended. All communications between nodes are handled transparently - regardless if these transmissions are within two nodes on the same or on different computers. Another advantage of this system is its very active community contributing to a large code base, including specific packages for marine robotics applications.

E. Navigation systems

The onboard navigation capabilities of the vehicle strongly depend on the sensors with which the vehicle is equipped. At the surface, estimation of the navigation data (i.e. position and velocity vector) is usually performed using GPS, since it provides high-accuracy position fixes at the very high rate of 10 Hz. Underwater, this problem becomes much more challenging. The vehicle can be equipped with a Doppler Velocity Log (DVL), which measures the velocity vector with respect to the water and to the bottom. If a DVL is not available, the



(a) Core equipment

(b) Optional equipment

Fig. 3. Equipment onboard the MEDUSA vehicles.

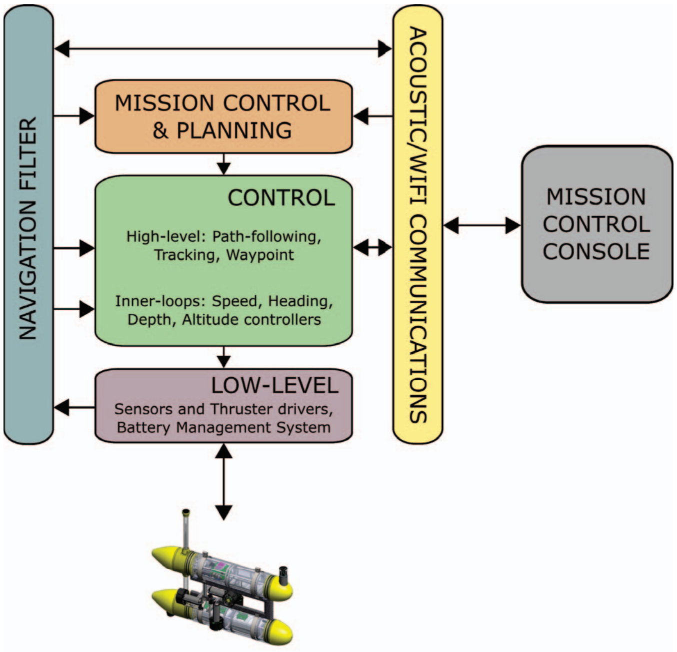


Fig. 4. Simplified diagram of the software architecture on the MEDUSAs.

RPMs of the thrusters can also be used to estimate the velocity of the vehicle through water using a simple model - at the cost of decreased accuracy. These velocity vectors can be rotated to an Earth-fixed frame by using an attitude estimate provided by an Inertial Measurement Unit (IMU). Integrating these velocity measurements provides position estimates over time (dead-reckoning), but leads to growing error. If accurate navigation is needed during a long period of time, then a sensor to provide some kind of position information should be available, such as an acoustic ranging device to provide distance measurements to a known agent, or an Ultra-Short Baseline (USBL) device to provide a position vector relative to a known agent. These agents could be autonomous vehicles themselves, opening the possibility of cooperation between multiple agents to improve navigation of one or more specific AUVs.

The information from these sensors needs to be properly integrated and filtered to provide estimates of position, velocity and velocity of the ocean current. This is performed by a navigation filter - an implementation of an Extended Kalman Filter based on a simple model, flexible enough to incorporate information from a wide variety of sensors. To simplify the navigation architecture, this filter estimates navigation information *on the horizontal plane only*: it is assumed that all vehicles carry a sensor capable of measuring depth independently, and that the motion in the vertical does not considerably influence the motion on the horizontal plane.

The continuous-time state space dynamic model for an AUV that can get navigation information from n other vehicles/nodes in a fleet is shown in eq. (1). The state $\boldsymbol{p}$ denotes the 2D position of the AUV in an Earth-fixed, North-oriented frame, $\boldsymbol{v}$ denotes its velocity vector, and $\boldsymbol{v}_c$ the velocity vector of the ocean current, which we assume to be constant to reflect the fact that, in practice, it is slowly-varying. States $\boldsymbol{p}_i$, $\boldsymbol{v}_i$ denote the 2D position and velocity vectors of node i of the fleet.

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{v}}_c \\ \dot{\boldsymbol{p}}_1 \\ \dot{\boldsymbol{v}}_1 \\ \vdots \\ \dot{\boldsymbol{p}}_n \\ \dot{\boldsymbol{v}}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{v}_1 \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{v}_n \\ \boldsymbol{0} \end{bmatrix} \tag{1}$$

The information is incorporated in the filter through system outputs, for a number of possible measurements a vehicle can acquire:

1) velocity with respect to the bottom given by DVL,

$$\boldsymbol{y}_1 = \boldsymbol{v}; \tag{2}$$

2) velocity with respect to the water given by DVL,

$$\boldsymbol{y}_2 = \boldsymbol{v} - \boldsymbol{v}_c; \tag{3}$$

3) velocity with respect to the water estimated by a model based on the thrusters RPMs,

$$\boldsymbol{y}_3 = \boldsymbol{v} - \boldsymbol{v}_c; \tag{4}$$

4) position given by GPS,

$$\boldsymbol{y}_4 = \boldsymbol{p}; \tag{5}$$

5) position of node i, possibly broadcast by node i using acoustic communications,

$$\boldsymbol{y}_5 = \boldsymbol{p}_i; \tag{6}$$

6) velocity of node i, possibly broadcast by node i using acoustic communications,

$$\boldsymbol{y}_6 = \boldsymbol{v}_i; \tag{7}$$

7) range to node i, obtained by an acoustic modem and transformed to the horizontal plane using information about the depth of node i,

$$\boldsymbol{y}_7 = \left\| \boldsymbol{p}_i - \boldsymbol{p} \right\|; \tag{8}$$

8) bearing (angle in the horizontal plane) to node i, obtained by a USBL,

$$\boldsymbol{y}_8 = \angle(\boldsymbol{p}_i - \boldsymbol{p}). \tag{9}$$

These measurements are not all mandatory. Observability of the position state $\boldsymbol{p}$ requires only either 1) GPS (from which the position is obtained directly), or 2) positions of at least two other nodes and the ranges to those nodes, or 3) the position of at least one other node and the range and bearing to that node. Still, underwater measurements related to position are usually obtained at slow rates (one measurement every few seconds), so velocity information (usually obtained at a much higher rate) is very helpful in improving the position estimate. Additionally, to obtain an estimate of the velocity of the ocean current, velocity measurements with respect to the water are needed (either from a model or, ideally, from a DVL).

The filter is equipped with mechanisms to incorporate measurements with delay by keeping a buffer in chronological order of the measurements obtained in a moving window time interval, along with the state estimate and its covariance after the incorporation. When an old measurement is received, it is inserted in the chronologically correct position in this buffer and all the computations are repeated from that point onwards until the current time. When incorporating a measurement, an outlier-test can be performed that computes the probability of obtaining the measurement given the current covariance matrix and rejects the measurement if this probability is below a certain threshold.

F. Control systems

As stated before, the MEDUSAs are equipped with several control systems that higher-level features can build upon. Most of these control loops were designed under the assumption that the maneuvers typically executed are smooth enough for the motions in surge, yaw and heave to be decoupled. Before testing in water, these controllers have all been tested using a dynamic model that approximates the true behavior of the vehicles, including the coupling between motions.

We now describe some of the lower-level control loops and some examples of control algorithms that can currently run on top of those.

1) Speed: The speed controller is responsible for keeping the vehicle at a desired longitudinal speed with respect to either the bottom or the water. Estimates of state to be regulated can be obtained from the navigation filter by projecting $\boldsymbol{v}$ (velocity with respect to the bottom) or $\boldsymbol{v} - \boldsymbol{v}_c$ (velocity with respect to the water) on the longitudinal axis of the vehicle. Regardless of which of these two velocity estimates is regulated, we now represent it as $\hat{u}$ and the reference to be tracked as u_d, and define the error to be driven to zero as

$$\tilde{u} = \hat{u} - u_d. \tag{10}$$

Finally, a Proportional-Integral (PI) control law computes the commanded *common mode* (CM) for the two longitudinal thrusters as

$$\text{CM} = -k_p \tilde{u} - k_i \int_0^t \tilde{u}(\tau) d\tau, \tag{11}$$

where the common mode is defined as the mean of the commands for the two thrusters, in percentage of the maximum RPMs:

$$\text{CM (\%)} = \frac{\text{Left RPMs (\%)} + \text{Right RPMs (\%)}}{2}. \tag{12}$$

2) Heading: The goal of the heading controller is to automatically steer the vehicle in a given desired direction ψ_d. We define a heading error

$$\tilde{\psi} = \psi - \psi_d \tag{13}$$

as the variable to be driven to zero, and apply the necessary transformations so that $\tilde{\psi} \in [-\pi, \pi[$. The commanded *differential mode* (DF) for the two longitudinal thrusters is then computed by a Proportional-Integral-Derivative (PID) controller through

$$DM = -k_p \tilde{\psi} - k_d \dot{\tilde{\psi}} - k_i \int_0^t \tilde{\psi}(\tau)\, d\tau, \tag{14}$$

where the error derivative $\dot{\tilde{\psi}} = \dot{\psi} - \dot{\psi}_d$ is computed using the yaw rate $\dot{\psi}$ measured by the Attitude and Heading Reference System (AHRS) and an estimate of the derivative of the heading reference signal ψ_d. The differential mode is defined by

$$\text{DM (\%)} = \frac{\text{Left RPMs (\%)} - \text{Right RPMs (\%)}}{2}. \tag{15}$$

The commands sent to each individual horizontal thruster are computed using eqs. (12) and (15). If any of these two controllers is disabled, the value of its corresponding mode (common for the speed, differential for the heading controller) is set to zero.

Alternatively to this heading controller, the yaw motion can also be controlled using a yaw-rate controller, which may be more convenient depending on the goal. The description of this controller is beyond the scope of this paper.

3) Depth and Altitude: A depth or altitude controller is responsible for issuing commands to the vertical thrusters so that the vehicle follows a reference signal of vertical distance to the surface (depth) or sea bottom (altitude). We denote a depth reference as z_d and an altitude reference as h_d. The vehicle uses a depth cell to measure z and an echo-sounder pointed vertically to measure h (with no preview in time). A depth error is computed using either

$$\tilde{z} = z - z_d \tag{16}$$

when the depth controller is active, or

$$\tilde{z} = h_d - h, \tag{17}$$

when the altitude controller is active. Then, a PID controller with feedforward acceleration produces the commands for the common mode of the vertical thrusters using the control law

$$\begin{aligned}\text{CM}_{\text{vertical}}(\%) = -k_p\tilde{z} - k_d\dot{\tilde{z}} - k_i \int_0^t \tilde{z}(\tau)d\tau \\ + \frac{\ddot{z}_d - \alpha\dot{z} - \beta\dot{z}|\dot{z}|}{\gamma},\end{aligned} \tag{18}$$

where the last term (feedforward) is related to the force necessary to overcome the vertical drag of the vehicle, based on a model.

4) Waypoint/Hold-position: Often, human operators need time to program missions and want the vehicles to stand by and not drift with ocean currents. It is also important to be able to send the vehicles to a specific point in space to collect some measurements for a predefined period. This motivated the development of a waypoint/hold-position algorithm to steer the vehicle to a given point $\boldsymbol{p}_d = (x_d, y_d)$ and keep it in a neighborhood of this point. The algorithm employs the following strategy (see [1] for details):

- issue a heading command in the direction towards the specified point and a speed command that decreases with the distance, computed by:

  $$u_d = k_u \sin^{-1}\left(\frac{d}{|d| + k_s}\right)\frac{2}{\pi} \tag{19}$$

  $$\psi_d = \angle(\boldsymbol{p}_d - \boldsymbol{p}) = \operatorname{atan2}(y_d - y, x_d - x), \tag{20}$$

  where

  $$d = \sqrt{(x - x_d)^2 + (y - y_d)^2} - \epsilon_d, \tag{21}$$

  $k_u > 0$ defines the upper limit on the speed reference u_d, and $k_s > 0$ is a parameter for tuning the vehicle speed with respect to distance error;
- when the vehicle reaches a neighborhood of radius ϵ_d of the desired point, set the speed reference to zero and maintain the previous heading.

5) Path-Following: The goal of the path-following algorithm is that the vehicle converges to and follows a path at a prescribed velocity profile. There is no timing law associated to this path, i.e. the algorithm does not define a desired position for the vehicle for each time instant. The current implementation of the path-following algorithm running in the MEDUSA vehicles works with paths defined in a horizontal plane, at a constant depth. A detailed description can be found in [2]; in short, the algorithm continuously:

- computes the position of the point target point $\boldsymbol{p}_c$, the point on the path closest to the estimated position of the vehicle $\boldsymbol{p}$;
- computes the angle β of the vector tangent to the path at $\boldsymbol{p}_c$, in the direction that the path should be followed;
- computes the cross-track error e as the projection of the position of the vehicle relative to the position of the closest point, $\boldsymbol{p} - \boldsymbol{p}_c$, on a direction $\beta + \frac{\pi}{2}$;
- issues a heading command, computed by a Proportional-Integral (PI) controller on e, and a speed command to the inner-loops, as given by

  $$\psi_d = \beta + \sin^{-1}\left[\sigma\left(-\frac{K_p}{u}e - \frac{K_i}{u}\int_0^t e(\tau)d\tau\right)\right] \tag{22}$$

  $$u_d = u(\boldsymbol{p}_c), \tag{23}$$

  where $\sigma(\cdot)$ is a saturation function and $u(\cdot)$ defines the desired velocity as a function of the position along the path (usually constant).

6) Cooperative Path-Following: The cooperative path-following (CPF) algorithm, as studied in [3], extends the path-following algorithm to a multiple-vehicle setting. The goal is that the vehicles follow their own paths in a coordinated fashion, i.e. are at corresponding positions on their paths at each instant in time. To simplify the notation, the paths to be followed are parametrized by an along-path coordinate γ. The value of this coordinate for the point on the path closest to vehicle i is denoted by γ_i. The coordination objective can now be stated as $\gamma_i = \gamma_j$ for any pair of vehicles (i, j). If the parametrizations are chosen appropriately, this objective corresponds to the vehicles maintaining a desired formation geometry. This coordination is achieved by having the vehicles exchange their own values of γ with their neighbors and then adding a correction to the speed command in eq. (23), yielding

$$u_d = u(\gamma) - k_{sync}\sin^{-1}\left(\frac{\tilde{\gamma}}{|\tilde{\gamma}| + k_s}\right), \tag{24}$$

where the coordination error $\tilde{\gamma}$ is defined for each vehicle i as

$$\tilde{\gamma}_i = \gamma_i - \frac{1}{|N_i|}\sum_{j \in N_i}\gamma_j. \tag{25}$$

The stability properties of this type of algorithms and the influence of the network topology (the definition of the set of neighbors for each vehicle) have been formally analyzed in

[3]; in practice, the set of neighbors usually contains all the remaining vehicles in the mission. This approach was applied in the scope of the Co3-AUVs project, described later in this paper.

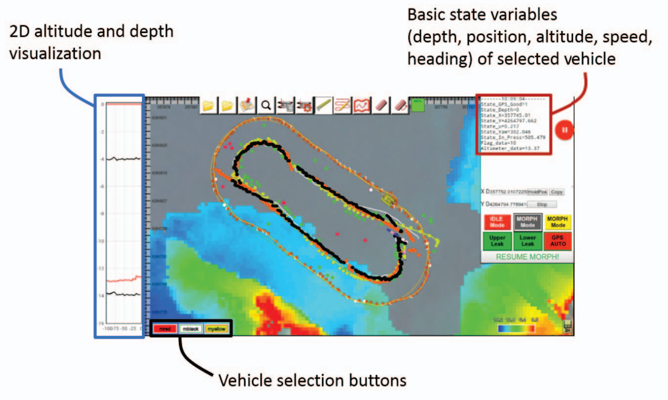
7) Tracking: In contrast to path-following, the tracking algorithm is used to make the vehicle track a desired trajectory *with an associated timing law* - the position where the vehicle should be is specified for each time instant. The controller makes the vehicle position $\boldsymbol{p}$ converge to and track the position of a virtual target $\boldsymbol{p}_d$, which moves according to the trajectory specification. Commands to the speed and heading inner-loops are computed in three steps:

1) compute the position error
$$\boldsymbol{e} = R^T(\psi)\left(\boldsymbol{p} - \boldsymbol{p}_d\right), \tag{26}$$
in the body-fixed frame of the vehicle, where the matrix $R^T(\psi)$ rotates a vector from the North-oriented frame to the body-fixed frame of the vehicle;
2) compute a desired velocity vector $\boldsymbol{v}_d$ with respect to the water using the control law
$$\boldsymbol{v}_d = \underbrace{\dot{\boldsymbol{p}}_d - \hat{\boldsymbol{v}}_c}_{\text{feedforward}} - \underbrace{R(\psi)K\sigma(\boldsymbol{e})}_{\text{saturated feedback}}, \tag{27}$$
where $\sigma(\cdot)$ is again a saturation function and K is a 2×2 gain matrix. The current implementation uses a component-wise scaled hyperbolic tangent as saturation function. In short, this control law includes a feedforward term to cancel the estimate of the current $\hat{\boldsymbol{v}}_c$ and the velocity of the target $\dot{\boldsymbol{p}}_d$, and a saturated feedback term based on the position error in the body-fixed frame, which is then rotated back to the North-oriented frame through $R(\psi)$;
3) issue heading and speed (with respect to water) commands based on the magnitude and angle of the desired velocity vector:
$$u_d = \|\boldsymbol{v}_d\| \tag{28}$$
$$\psi_d = \angle\boldsymbol{v}_d. \tag{29}$$

In contrast with path-following, the tracking algorithm does not need full a-priori information of the trajectory to be tracked; it requires only that the vehicle can compute the position and velocity vectors of the target at the current time, $\boldsymbol{p}_d$ and $\dot{\boldsymbol{p}}_d$. This makes it easy to adapt this algorithm for multiple-vehicle formation control, by generating the targets for each vehicle such that they comply with the desired formation geometry. This approach was used in the scope of the MORPH project, to be described later; see [4], [5] for details.

G. Mission control console

Operations with multiple MEDUSA vehicles are monitored from a control room onshore. The information flowing to and from the vehicles, including commands sent from the control room and status monitoring data sent from the MEDUSAs, are managed by a mission control console. This software runs on a web browser for interoperability, in computers connected to the vehicle's WiFi network via a switch. The graphic interface is gradually being adapted to comply with Google's Material Design guidelines, in order to make the console look and feel more uniform.



```
-------10:09:04-------
State_GPS_Good=1
State_Depth=0
State_X=357745.01
State_Y=4264797.662
State_u=0.217
State_Yaw=302.046
State_In_Press=505.479
Flag_data=10
Altimeter_data=13.37
```

Fig. 5. Front screen of the console and main panels.

We now describe the user interface and main features of this software, developed in-house at IST. Two main functions of the console are described: mission monitoring (the console displays information about the status of the vehicles) and mission programming (the user sends commands to the vehicle, such as 2D trajectories to be followed).

1) Mission Monitoring: Figure 5 shows the main screen of the console and the panels available. The background screen is a top-view of the mission area, optionally including texture from a map or a bathymetric color map loaded from files. Positions obtained from the vehicles via ROS messages can be configured to show on the screen by selecting the ROS topic name, the color and style of the representation (dotted or continuous), and optionally a vehicle icon.

The vehicle selection buttons at the bottom allow the operator to select a specific vehicle by clicking on its ID. The panel on the right displays basic state variables concerning the selected vehicle, such as depth, position, altitude, speed and heading. If the vehicles are at the surface, all this information is obtained via WiFi. When a vehicle is underwater, this data will be unavailable. Alternatively, this information can be i) obtained through a buoy towed by the vehicle, with a WiFi antenna at the surface (for limited depths only), or ii) broadcast to a surface vehicle via acoustic communications, and then relayed to the mission control room via WiFi. For each of the active vehicles, the panel on the left plots two lines as a function of time for the last 100 seconds: the depth and total water column.

2) Mission Programming: The console is also used to send commands to the vehicles. Besides simple commands, such as waypoint/hold position and constant depth references, complex trajectories can be designed graphically and sent to the vehicles to be followed either independently or in a synchronized fashion using the CPF algorithm. These more complex trajectories are based on lines, arcs, lawnmowers,

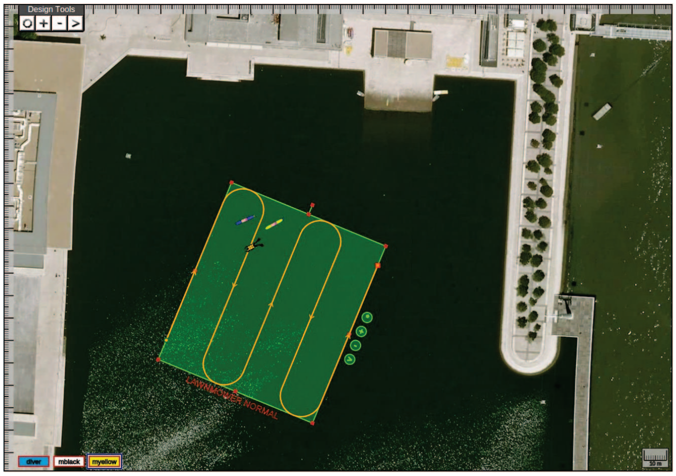


Fig. 6. Mission programming: drawing a lawnmower mission.

zero-shapes, circles, star-shapes and eight-shapes (these last two are particularly useful for sensor calibration). Using only the graphic interface, the user can resize, rotate, apply offsets, add or reduce the number of legs in lawnmower shapes, and other basic customization operations. The trajectories can then be saved to files and uploaded to the vehicles. Figure 6 shows the mission programming interface for lawnmower shapes.

IV. APPLICATIONS IN EU PROJECTS

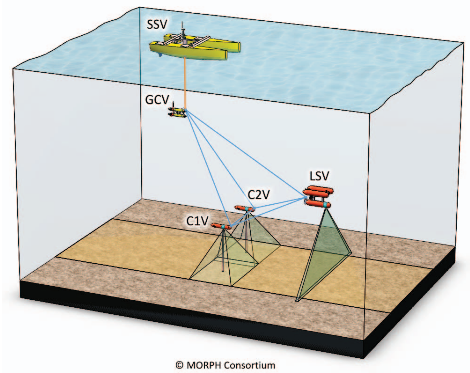
Over the past seven years, the MEDUSA vehicles have played an increasingly important role in the scope of several EC funded projects. In what follows we focus on the most representative ones.

A. *Co3-AUVs, 2008-2011*

Co3-AUVs (cooperative cognitive control for autonomous underwater vehicles [6]) was a 3-year project with the participation of three academic and one industrial partner. The project proposed to develop, implement and test cognitive systems for coordination and cooperative control of multiple AMVs. Under this general scope, aspects investigated included 3D perception and mapping, cooperative situation awareness, deliberation, and navigation. The demonstrated end results were a harbor scenario test and human diver assistance scenario involving a fleet of AMVs. Three surface MEDUSAS were involved in the project as part of the fleet.

B. *MORPH, 2011-2015*

MORPH (marine robotic system of self-organizing, logically linked physical nodes [7]) was a 4-year project bringing together a consortium involving multiple partners from both the industry and academia. The key idea in the project is the use of a flexible, environment-compliant formation of 5 autonomous vehicles with different roles and complementary capabilities to enable mapping of underwater environments near challenging scenarios such as vertical cliffs. The idea of resorting to a fleet of such vehicles stems from the realization that the traditional approach to underwater mapping involves the use of a single, heavy, bulky AUVs equipped with a number of very expensive sensors - imposing heavy costs and risk on those operating such vehicles. The use of a fleet allows each of the vehicles to be much smaller, since no single vehicle needs to be equipped with all the necessary sensors, while reducing the risk of the operation by distributing expensive equipment through multiple nodes.



© MORPH Consortium

Fig. 7. Schematic drawing of the MORPH formation flying over flat terrain.

IST participated in this project both as a vehicle provider and as responsible for developing most of the cooperative control and navigation algorithms. Overall, three MEDUSA-class vehicles participated in the trials at sea. Advances in the study of cooperative control and navigation algorithms for the specific MORPH application were reported in a series of publications [4], [5], [8], [9].

C. *CADDY, 2014-2016*

CADDY (cognitive autonomous diving buddy [10]) is an ongoing 3-year project motivated by the fact that divers operate in harsh conditions where any problem can have catastrophic consequences, while carrying heavy equipment in complex environments. The goal of CADDY is to develop and test a system of two autonomous marine vehicles to monitor and help human divers accomplish inspection tasks underwater in a more productive and safer manner - one at the surface and one underwater, as illustrated in fig. 8. These robots exhibit cognitive capabilities, being able to see, learn, and adapt to the diver's behaviour.

IST participates in this project as vehicle provider - the main surface vehicle and one of the spare underwater vehicles are MEDUSAS - and one of the partners responsible for control and navigation of both vehicles. Work undergoing or achieved by IST so far includes:

- a tracking algorithm based on artificial potential techniques for the surface vehicle, which should remain close to both the diver and the underwater vehicle to ensure proper function of the acoustic network, while strictly avoiding being directly on top of any of these;
- a navigation filter running on the underwater vehicle, akin to the one described in section III-E;
- a path-following algorithm to be used on the underwater vehicle, so that the diver can command the vehicle to

Fig. 8. An illustration of the concept behind the CADDY project.

follow a specific trajectory while acquiring imagery to generate a mosaic (one of a set of commands that should be available to the diver under a mode named BUDDY Slave);

- a trajectory generator for the surface vehicle to localize both underwater nodes based only on range measurements in an optimal manner.

D. *WiMUST, 2015-2017*

WiMUST (widely scalable mobile underwater sonar technology [11]) is an on-going 3-year project that started in early 2015, involving 4 academic and 5 industrial partners. The goal of the consortium is to expand and improve the functionalities of current cooperative marine robotic systems by designing a scalable system of autonomous underwater vehicles to enable acoustic array technologies to be employed in a distributed fashion. Geotechnical survey operations require precise geometries to be maintained between an acoustic source and acoustic receivers, usually placed along streamers. These surveys are currently based on ship-towed streamers. This solution can easily get operationally cumbersome and its performance significantly affected by currents as the number and length of the streamers increases. The use of the proposed system to tow these streamers holds potential benefit both by simplifying and automating the operations required to maintain the necessary geometry (through geometric formation control of the AUVs) and by allowing the geometry of the formation itself to be adapted online, including the depth, putting forward the concept of an adaptive acoustic antenna for marine geotechnical surveying.

IST is involved in WiMUST as vehicle provider, through the expected use of 5 MEDUSA vehicles in the final formation, and as the partner responsible for the cooperative motion control algorithms. Secondary roles include participation in subjects such as software and hardware integration and acoustic communications. During the first year of project, IST has put an effort on getting insight into the dynamic behaviour of the streamers when towed by MEDUSA vehicles. Work is ongoing to identify the parameters of a parametric coupled streamer-vehicle system (CSVS). Other subjects under study at IST are the underwater acoustic communication and navigation architecture and the application of cooperative motion strategies to the WiMUST scenario.

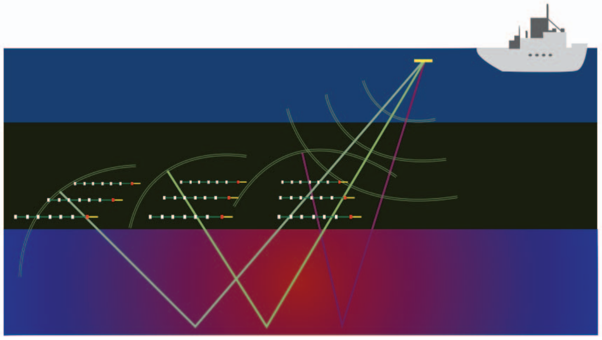


Fig. 9. A schematic illustration of the technology to be developed under WiMUST.

E. MEDUSA *Deep Sea*

Building on IST's know-how obtained from years of developing and testing multiple-vehicle systems, the MEDUSA Deep Sea is an EEA-funded project that brings together partners from Portugal and Norway, aiming at extending the operational capabilities of the MEDUSA platform to depths up to 3000 m. The proposed solution is a system of multiple autonomous vehicles, involving a MEDUSA$_S$ ASV and one or more AUVs. One of the AUVs will be a new MEDUSA-class vehicle, MEDUSA$_{DS}$, with the required depth and autonomy capabilities for a 3000 m deep operational scenario. The development of this vehicle is currently on-going and the final demonstration of the system is scheduled for March 2017.

V. CONCLUSION

This paper gave a high-level overview of the MEDUSA vehicles and their design process. We started by stating a number of desired characteristics that motivated and constrained the vehicle design. A description of the vehicles followed, including mechanical and electrical design considerations and how they relate with the intended goals, an illustration of the software architecture, and a summary of the navigation system, lower-level control algorithms, and multiple-vehicle cooperative control strategies. The mission control console through which the operator interacts with the vehicles was also described, illustrating the tools available for mission monitoring and programming. Five past and ongoing European projects were introduced in which the MEDUSAs and IST play an instrumental role, and the main advances achieved so far under each of these projects were described. The increasing robustness and reliability of these vehicles since they were first developed has greatly contributed to the European pool of marine vehicles available for marine robotics research.

ACKNOWLEDGMENT

This work was partially supported by the EC projects MORPH (FP7 GA No. 288704), CADDY (FP7/ICT/2013/10), and WiMUST (GA No. 645141), and the Portuguese FCT Funding Agency [PEst-OE/EEI/LA0009/2011]. We are extremely grateful to all the members of ISR/IST and everyone involved in the Co3-AUVs, MORPH, CADDY, WiMUST, and MEDUSA Deep Sea projects for their effort and persistence in this exciting journey of developing, implementing, and sea testing a number of core systems for cooperative mission programming, communications, control, navigation, and mapping. We are especially indebted to all partner institutions outside IST for their commitment and the fantastic cooperation environment that they helped create. We trust that this collaborative effort will continue to lead to successful results and technological advances in the marine robotics research ecosystem.

REFERENCES

[1] J. Ribeiro, “Motion control of single and multiple autonomous marine vehicles,” Master’s thesis, Instituto Superior Técnico, Lisbon, Portugal, 2011.

[2] P. Maurya, A. Aguiar, and A. M. Pascoal, “Marine vehicle path following using inner-outer loop control.” in *8th IFAC International Conference on Manoeuvring and Control of Marine Craft 2009, September 16-18, 2009, Guaruja-Brazil. 6 pp.* IFAC International Conference, 2009.

[3] R. Ghabcheloo, A. P. Aguiar, A. Pascoal, C. Silvestre, I. Kaminer, and J. Hespanha, “Coordinated path-following in the presence of communication losses and time delays,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 234–265, 2009.

[4] P. C. Abreu, M. Bayat, J. Botelho, P. Gois, J. Gomes, A. Pascoal, J. Ribeiro, M. Ribeiro, M. Rufino, L. Sebastiao, and H. Silva, “Cooperative formation control in the scope of the EC MORPH project: Theory and experiments,” in *OCEANS 2015 - Genova*, May 2015, pp. 1–7.

[5] P. C. Abreu, M. Bayat, A. M. Pascoal, J. Botelho, P. Góis, J. Ribeiro, M. Ribeiro, M. Rufino, L. Sebastião, and H. Silva, “Formation control in the scope of the MORPH project. Part II: Implementation and results,” in *Proc. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Girona, Spain, 28-30 April 2015.

[6] A. Birk, G. Antonelli, A. Caiti, G. Casalino, G. Indiveri, A. Pascoal, and A. Caffaz, “The CO3AUVs (Cooperative Cognitive Control for Autonomous Underwater Vehicles) project: Overview and current progresses,” in *OCEANS, 2011 IEEE - Spain*, June 2011, pp. 1–10.

[7] J. Kalwa, A. Pascoal, P. Ridao, A. Birk, M. Eichhorn, L. Brignone, M. Caccia, J. Alvez, and R. Santos, “The European R&D-project MORPH: Marine robotic systems of self-organizing, logically linked physical nodes,” in *Proc. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV2012*, vol. 3, no. 1, Porto, Portugal, 10-12 April 2012, pp. 349–354.

[8] F. Rego, J. M. Soares, A. Pascoal, A. P. Aguiar, and C. Jones, “Flexible triangular formation keeping of marine robotic vehicles using range measurements.” in *Proc. 19th IFAC World Congress*, Cape Town, South Africa, 24-29 August 2014.

[9] P. C. Abreu and A. M. Pascoal, “Formation control in the scope of the MORPH project. Part I: Theoretical foundations,” in *Proc. 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles*, Girona, Spain, 28-30 April 2015.

[10] N. Miskovic, M. Bibuli, A. Birk, M. Caccia, M. Egi, K. Grammer, A. Marroni, J. Neasham, A. Pascoal, A. Vasilijevic, and Z. Vukic, “Overview of the FP7 project “CADDY - cognitive autonomous diving buddy”,” in *OCEANS 2015 - Genova*, May 2015, pp. 1–5.

[11] H. Al-Khatib, G. Antonelli, A. Caffaz, A. Caiti, G. Casalino, I. Bielic de Jong, H. Duarte, G. Indiveri, S. Jesus, K. Kebkal, A. Pascoal, and D. Polani, “The widely scalable mobile underwater sonar technology (WiMUST) project: An overview,” in *OCEANS 2015 - Genova*, May 2015, pp. 1–5.