Sparus II AUV—A Hovering Vehicle for Seabed Inspection

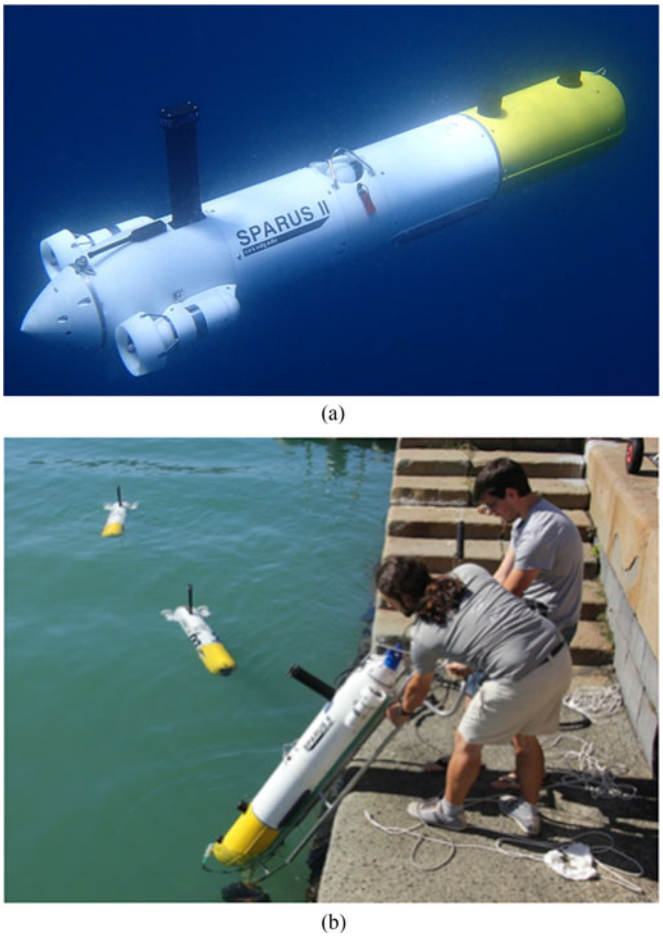
Marc Carreras, *Member, IEEE*, Juan David Hernández, *Member, IEEE*, Eduard Vidal, Narcís Palomeras, *Member, IEEE*, David Ribas, and Pere Ridao, *Member, IEEE*

***Abstract*—This paper proposes the use of path-planning algorithms for hovering autonomous underwater vehicles (AUVs) in applications where the robot needs to adapt online its trajectory for inspection or safety purposes. In particular, it proposes the platform Sparus II AUV and a set of planning algorithms to conduct these new AUV capabilities. These algorithms generate trajectories under motion constraints, which can be followed without deviations, to ensure the safety even when passing close to obstacles. View planning algorithms are also combined to decide the movements to be executed to discover the unexplored seabed or target, and to cover it with a camera or sonar. Online mapping with profiling sonars and online planning with fast sampling-based algorithms allow the execution of missions without any previous knowledge of the 3-D shape of the environment. Real 2-D results in an artificial harbor structure and simulated natural rocky canyon demonstrate the feasibility of the approach for avoiding or inspecting the underwater environment. These new AUV capabilities can be used to acquire images of the environment that can be used to inspect and map the habitat.**

***Index Terms*—Autonomous underwater vehicle (AUV), hovering AUV, online path planning and view planning (VP).**

I. Introduction

COMMERCIAL autonomous underwater vehicles (AUVs) are mainly conceived to surveying applications in which large areas must be covered and the vehicle follows safe paths at safe altitudes. New advances in sonar technology, image processing, mapping, and robotics will allow more complex missions, in which the AUV will be able to navigate at a closer distance from the seabed, it will react to the 3-D shape of the environment, and it will even perform some autonomous intervention tasks. In this context, the Underwater Robotics Research Centre of the University of Girona has been developing several AUV prototypes during more than 20 years to achieve these new capabilities. The Sparus II AUV [1] [see Fig. 1(a)] is one of them, and was conceived as a hovering AUV for surveying and inspection applications. The vehicle was developed in 2013 and during four years, many experiments have been carried out, from classical multibeam sonar and photomosaicing surveys to real-time mapping and motion planning between obstacles for seabed coverage and inspection. The AUV has also participated in several robotic competitions and has been replicated five times for external research institutions [see Fig. 1(b)].


(a)

(b)

Fig. 1. Sparus II AUV at sea. (a) Underwater view. (b) Three units participating at euRathlon competition in 2014.

Sparus II AUV is a lightweight hovering vehicle with mission-specific payload area and efficient hydrodynamics for long autonomy in shallow water (200 m). It combines torpedo-shape performance with hovering capability. It is easy to deploy and to operate. The payload area can be customized by the enduser and it uses an open software architecture, based on a robot operating system (ROS) [2], for mission programming. Its flexibility, easy

Manuscript received March 26, 2016; revised November 26, 2017; accepted January 7, 2018. This work was supported in part by the EXCELLABUST and ARCHROV Projects under Grant H2020-TWINN-2015, CSA, ID: 691980 and Grant DPI2014-57746-C3-3-R, respectively. The work of E. Vidal was supported by the Spanish Government under Ph.D. Grant FPU14/05493. *(Corresponding author: Marc Carreras.)*

Guest Editor: W. Kirkwood.

The authors are with the Computer Vision and Robotics Institute, Universitat de Girona, Girona 17003, Spain (e-mail: marc.carreras@udg.edu; juandhv@gmail.com; rc4559@gmail.com; narcispr@gmail.com; dribas@udg.edu; pere.ridao@udg.edu).

Digital Object Identifier 10.1109/JOE.2018.2792278

operation, and openness make the Sparus II AUV a multipurpose platform that can adapt to industrial, scientific, and academic applications. The following are the key features of the vehicle:

1) efficient hydrodynamics and long endurance due to its torpedo shape;
2) hovering capability which extends the maneuverability of classical AUVs;
3) lightweight vehicle, similar weight, and size than underwater gliders;
4) easy operation, which can be operated by two persons from any small boat;
5) mission-specific payload: open hardware for equipment integration;
6) software architecture based on ROS: open software available for download.

Classical AUV applications are generally conducted in a previously explored area so that the vehicle can navigate at a constant and safe altitude from the seafloor. In their simplest form, the AUV follows a sequence of precalculated waypoints to collect data, which is retrieved, processed, and analyzed after concluding the mission. Nonetheless, more recent applications seek to inspect in close proximity different kinds of structures such as in-water ship hulls and natural formations on the seafloor. To do this, a common characteristic is the necessity of *a priori* information of the area or structure to be inspected, either to navigate at a safe and conservative distance [3] or to precalculate a survey path that may be corrected or reshaped online [4]. In this respect, the University of Girona presented a 2.5-D coverage path-planning approach that allows an AUV to inspect complex structures such as an underwater boulder. To do so, the proposed framework plans offline a nominal coverage path using preliminary information (e.g., structure position and shape), and then it uses the robot navigation estimation and perception sensor uncertainties to adapt and reshape online the path according to the structure shape perceived while conducting the mission [5].

However, there are similar or even potentially new applications, where preliminary information might not be available. In these scenarios, AUV must operate in unexplored, cluttered, and dynamic environments, and therefore, it is more exposed to collisions. One option to deal with such constraints is to use a planner capable of planning the collision-free paths online, thus allowing to adapt and replan to overcome global position inaccuracy, especially when navigating in close proximity to nearby obstacles. In this respect, Petillot *et al.* [6] proposed a first approach for underwater vehicles to plan paths while avoiding obstacles online, which used real-world multibeam sonar data sets of acoustic images obtained by a remotely operated vehicle. However, they validated their approach by guiding a simulated model. Moreover, capability for mapping and planning online and simultaneously was not proven. Along this line, Maki *et al.* [7] presented a method to plan paths online, which used landmarks to guide an AUV. However, their approach did not permit replanning maneuvers and results were obtained in a water tank, i.e., in a highly controlled environment.

To overcome the aforementioned approaches limitations and to inspect the seabed with Sparus II AUV, we have worked with path-planning algorithms for adapting online the trajectory.

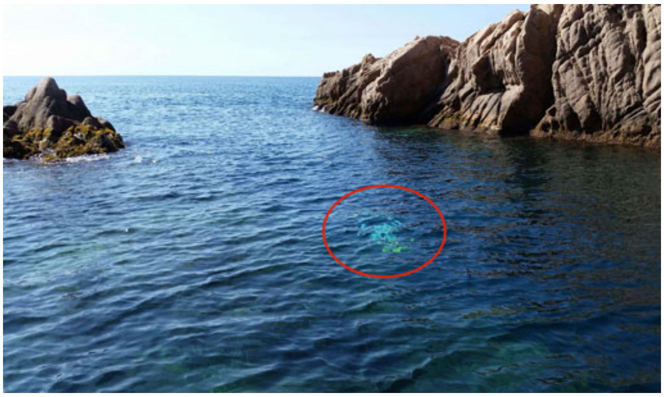


Fig. 2. Sparus II AUV conducting an autonomous mission that requires navigating through a rocky formation without a preliminary map.

We have developed a framework that endows an AUV with the capability to autonomously navigate environments for which no previous information is available. The framework consists of two main functional pipelines: 1) one that computes collision-free paths while simultaneously mapping the surroundings incrementally using a profiling sonar; 2) another that allows optical inspection of an underwater structure by online generating appropriate viewpoints at each iteration. The whole framework has been extensively evaluated in simulated and real-world scenarios using Sparus II AUV (see Fig. 2). Results have demonstrated the suitability of our approach for the aforementioned applications.

This paper proposes Sparus II AUV, a hovering capable vehicle, as an AUV able to extend the range of classical AUV applications, by real-time mapping, planning, and executing trajectories. This paper extends the work presented in [8] including a detailed description of Sparus II AUV, and point out its suitability for online trajectory planning. This paper is organized in three main sections. Section II details the AUV. Section III presents the path-planning approach. Section IV proposes a view planning (VP) paradigm to inspect unknown objects. Sections III and IV also include the real and simulated experiments pointing out the suitability of the algorithms and the AUV. Section V concludes this paper.

II. Sparus II AUV

A. Autonomous Underwater Vehicle Configuration

Sparus II AUV (see Fig. 3) has a torpedo-like shape to be efficient when navigating at medium/high velocities. The maximum velocity in surge is 4 kn. It has three thrusters (two horizontal and one vertical) that allow the control of the surge, heave, and yaw degrees of freedom (DOFs) when moving in hovering mode. In torpedo-based mode, two fins behind the horizontal thrusters are used for controlling the pitch DOF and, thus, controlling the depth or altitude of the vehicle. The fins have been integrated in a second phase, showing very good performance for stabilization at low and high surge velocities (see Fig. 4). The differential movement of the fins is also used to stabilize the roll DOF. Therefore, the robot is designed to be efficient when moving fast using the two horizontal thrusters and the two

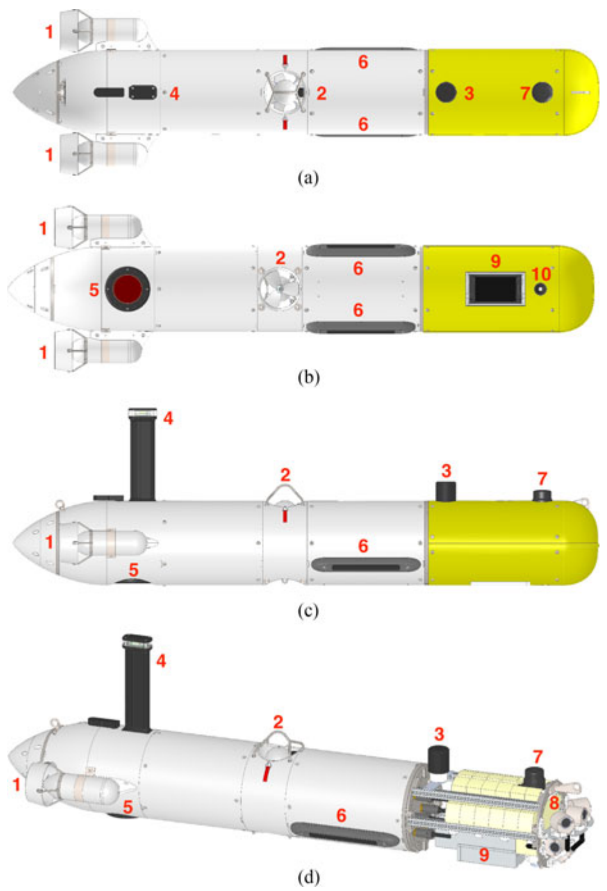


Fig. 3. Sparus II AUV views. Different hardware parts can be observed, including the thrusters (1,2), the acoustic modem (3), the Wi-Fi and GPS (4), as well as interoceptive and exteroceptive sensors such as the DVL (5), sidescan sonar (6), mechanically scanning imaging sonar (7), single-beam echosounders (8), multibeam sonar (9), and an optical camera (10). (a) Top view. (b) Bottom view. (c) Lateral view. (d) Three-dimensional view.

fins, for controlling surge, pitch, and yaw DOFs. The vertical thruster is not used in this mode, since the fins are able to control the depth of the vehicle more efficiently. Then, the robot can keep position or can move slowly using the vertical thruster for counteracting the buoyancy and floatation forces, thus acting as a normal hovering unmanned underwater vehicle (UUV). It is interesting to note that, in hovering mode, the two fins will still be active for having a zero pitch and roll position, and this stabilizes the vehicle, especially in heave and surge movements. Finally, it is worth noting that the vehicle does not have the sway DOF, which would be necessary for a complete hovering AUV. However, this partially hovering configuration is usually enough for most applications and keeps the simplicity of the vehicle.

B. Mechanical Structure and Electronics

The hull has been designed having a main closed housing and an open payload area in front. The maximum depth of the vehicle is 200 m, for which aluminum and acetal were chosen as construction material. The front payload area is shown in yellow. The main housing is represented in white. This housing is assembled with two cylinders, a special part for the vertical thruster and another special part for the back cone, where horizontal thrusters, Doppler velocity log (DVL), and antenna are placed. Two endcups are used in front and on the back for closing the main hull, having some underwater connectors for different purposes. Another special part is attached to the back endcup, which contains the two servomotors for the fins. The main idea behind the design of the hull is that the internal electronics is attached to the central part, the one containing the vertical thruster. When disconnecting the front endcup and the back cone part, the two cylinders can be removed and all the electronics is accessible. The electronics and all the cables do not have to be moved or pulled to open the vehicle, and this improves a lot the robustness of the system. The only thing that has to be done, before removing the two cylinders, is to disconnect three multipin industrial connectors. It is possible to connect them again without the cylinders for testing the electronics with all systems connected.

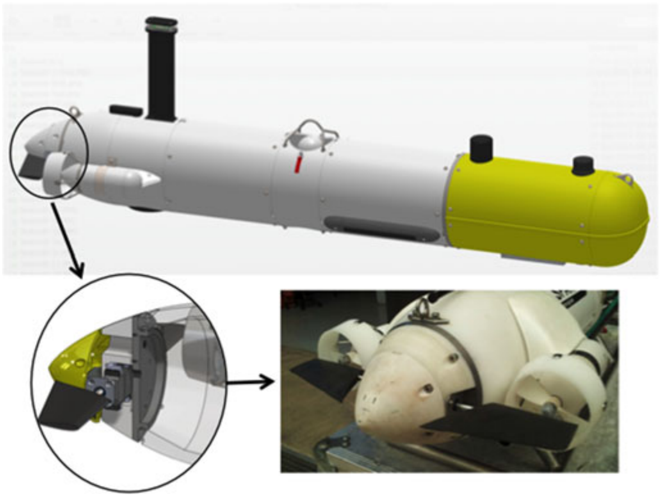


Fig. 4. Horizontal fins for vehicle stabilization and pitch control.

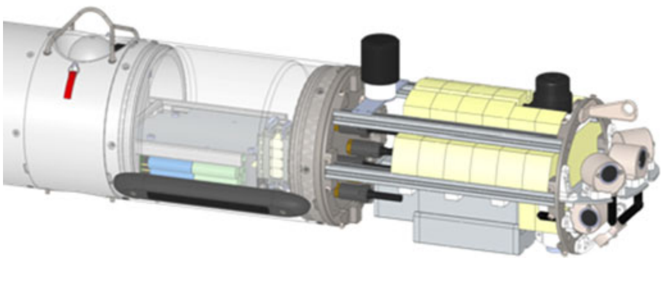


Fig. 5. Internal and external payload area, containing an aluminum structure, foam, two sensors, and underwater connectors.

The payload area has been designed to be flexible to different equipment requirements (see Fig. 5). It has an internal part, for additional electronics or computer, and the external part. To integrate the equipment, an external structure is used, together with the corresponding foam, weight, and the external yellow skin, which is built on a mold with acrylonitrile butadiene styrene (ABS) plastic. For each equipment configuration, a new skin can be adapted by cutting the plastic with the required apertures. The payload area can integrate any equipment having a maximum volume of 8 L and a maximum weight of 7 kg. The nonrequired volume or weight is filled by foam and lead to maintain a constant total volume and weight, which is approximately 52 L and 52 kg in air. Both the size and weight are

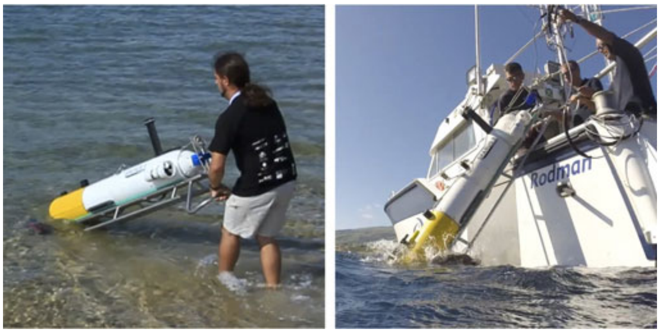


Fig. 6. AUV deployment from the shore and from a boat.

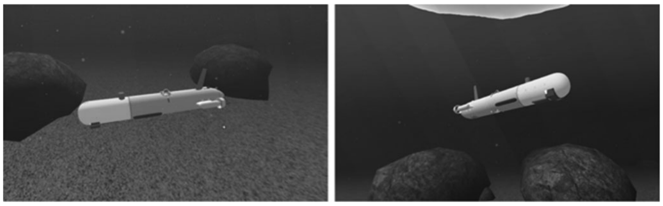


Fig. 7. Views of Sparus II simulated in UWSim.

similar to conventional gliders and, therefore, the deployment and recovery of the vehicle can be done by a team of two persons from a small boat without requiring a crane. A trolley has been designed for easy deployment, recovery, and transportation; see Fig. 6. After finishing the development of the vehicle, a set of experiments were conducted to estimate the hydrodynamic model of the vehicle. The Appendix provides the equations of the model, which follows the convention established by Fossen [9], including the experimentally identified parameters.

The vehicle is powered by a 1.4-kWh pack of Li-ion batteries, which is controlled by a battery management system (BMS) to ensure the safety of the pack when charging and discharging. All cells are monitored by the BMS and the onboard computer receives the state of the pack continuously. The estimated endurance of the batteries, depending on the velocity of the vehicle and the consumption of the payload, is between 8 and 12 h. At 1 kn of surge speed, without payload power consumption, the required electrical power is around 100 W, which would allow a 14-h operation.

C. Equipment and Software Architecture

The robot has a PC104 embedded computer that manages all systems. The computer runs an Ubuntu Linux distribution with the ROS open framework, which standardizes the integration of new devices or systems and opens the software to any user. An open-source control architecture called COLA2 [10], developed by the University of Girona, is managing all equipment and systems of the vehicle. The architecture can be tested in simulation, before real experiments, with the UWSim underwater simulator [11] (see Fig. 7), which considers the hydrodynamic model of the vehicle (see the Appendix). The computer uses an RS485 serial communication for controlling the three thrusters, which have been designed also by the University of Girona. Each thruster provides up to 7 kg of force using a brushless direct current (dc) motor with a magnetic coupled ducted propeller. The same serial line is used to control the two fins using servomotors connected by means of a sealed axis. In case of a water leakage, the servomotors are contained in an independent part, which is not connected internally to the main hull housing. For navigation, the robot uses an inertial measurement unit (IMU) from Analog Devices, Norwood, MA, USA, a standard GPS, a pressure sensor from Keller, London, U.K., and an original equipment manufacturer (OEM) DVL from Teledyne RDI, Poway, CA, USA, which is integrated in the back cone part. An OEM acoustic modem with ultrashort baseline (USBL) from Evologics is integrated in the payload area. All the systems are switched ON by a magnetic switch and by a remote controller. The internal computer can be connected by Wi-Fi or using an umbilical cable. Application-specific equipment is installed in the payload area, having also space inside the main hull for internal electronics. Regulated 12 V and 24 V and Ethernet and RS232 serial communication is available for new equipment. Examples of sensors that can be integrated are: Imagenex multibeam profiler sonar, Point Grey BumbleBee stereo camera, Seabird CTD, Tritech mechanically scanned acoustic profiler for obstacle avoidance, and Soundmetrics ARIS forward looking sonar.

The navigation system consists of an extended Kalman filter, which uses all information available from the sensors. This filter estimates the vehicle position and velocity at any given time instance. Position estimates are computed in the global reference frame in North–East–Down coordinates, and the velocity estimates are computed in the body reference frame. The filter is first initialized with the GPS. A position update to the filter is done when valid GPS or pressure sensor measurements are available. It is also possible to perform position navigation updates using USBL information, taking into consideration the time delay in which the USBL data were received. Similarly, when good DVL readings are available, a velocity update to the filter is applied. No probabilistic approach is used regarding vehicle orientation in the navigation filter, although an automatic calibration procedure is done to perform hard iron and soft iron corrections to the magnetometer, and proper gyroscope bias estimation is also computed.

The control system of the Sparus II AUV has been designed to manage a total of 5 DOFs (surge, heave, roll, pitch, and yaw) from the robot's actuators (three thrusters and two fins). It contains a high-level controller, which is responsible for generating some guidance methods such as the line of sight (LOS) and strategies to keep one fixed position or velocity. It also has a safety node, which continuously supervises the hardware state and checks if the robot's location accomplishes the safety parameters defined as the maximum depth and minimum altitude. Finally, a low-level controller is set in three hierarchical levels corresponding to the position, the velocity, and the force control. The position and velocity controls are composed by a filter, whose function is to merge the different set-points that can be requested at the same time taking into account their priority, and either or both of a model or a PID controller, whose output becomes the input of the next control. After that and once all force requests have been merged, the desired one for each DOF is obtained. These forces are combined considering the

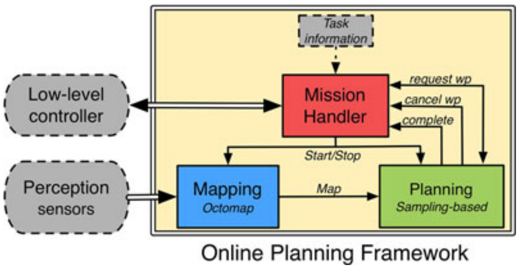


Fig. 8. Online path-planning framework and its main functional modules.

actuators' location and finally converted to actuators' set-points by a dynamic model.

Although the majority of DOFs are computed individually from the top to the bottom of the control system, the desired heave pose is reached by combining a force on heave and a torque on pitch. A set of fuzzy conditions, which depend on the robot's depth and surge velocity, determine how the vertical thruster and fins should work. For instance, at low surge speed, the depth positions are reached just with the vertical thruster, while at high speed they are achieved only with the fins; between these two extremes, the vertical thruster and the fins functionality are gradually adapted to the depth and surge velocity feedback through different states, avoiding sharply changes in the robot's behavior.

III. Online Path Planning

To successfully navigate in unexplored environments, the high-level controller has a framework which includes a functional pipeline that is in charge of simultaneously mapping the surroundings and planning motions for the AUV [12]. This motion planning pipeline is composed of three main modules: a mapping module that incrementally builds an occupancy map using Octomaps [13], a planning module that generates online collision-free motions, and a mission handler module that works as a high-level coordinator of the planner and the AUV controllers (see Fig. 8).

A. Mapping Module

The mapping module incrementally builds a representation of the environment by using data received from a mechanically scanned profiling sonar, which provides a range of information about nearby obstacles that, combined with the vehicle's navigation (position and orientation), permits establishing the free, occupied, and unexplored space with respect to an inertial coordinate frame. To represent this data, we use an octree-based framework called Octomap [13].

B. Planning Module

The planning module uses the asymptotic optimal RRT (RRT*) [14], a sampling-based algorithm that calculates a collision-free path from the current vehicle position to a specified goal. This path-planning algorithm, as well as other variants of the rapidly exploring random tree (RRT) [15], builds a tree of collision-free vehicle configurations. However, given that the environment is initially unexplored for the proposed applications, we have modified the RRT* to continuously check and reshape (replan) the path according to the surroundings information that is gathered as the vehicle moves toward the goal.

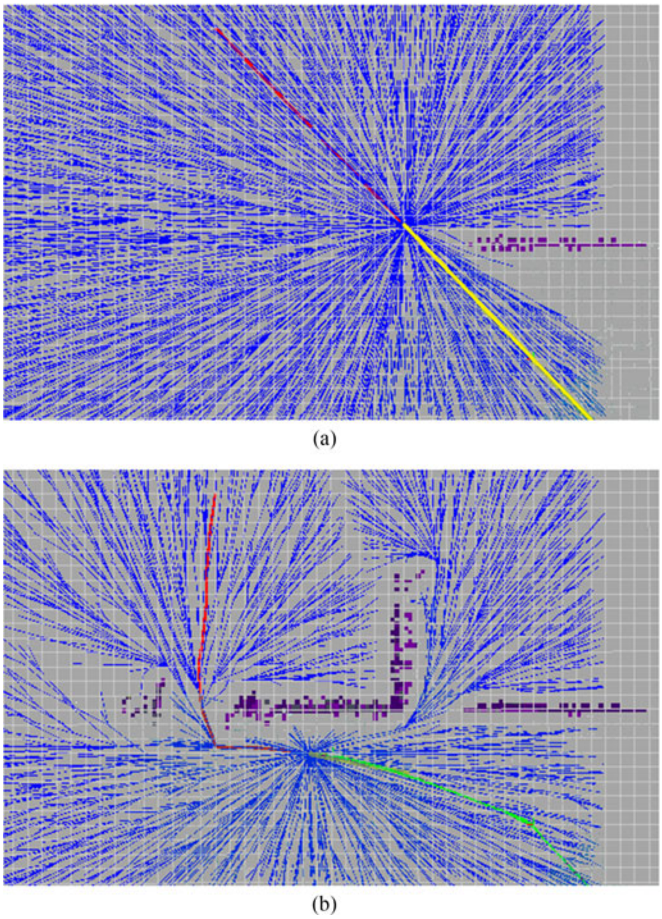


Fig. 9. Sparus II AUV conducting an autonomous mission in a simulated scenario. As the vehicle follows the path (red), the modified RRT* discards those branches that result under collision (with nearby obstacles—in purple) when the corresponding region has been explored.

Fig. 9 depicts a simulation of the Sparus II AUV conducting an autonomous mission in an unexplored environment, in which the vehicle is guided by our modified RRT*. However, in this first approach, the path planner does not consider the vehicle motion constraints, which is especially critical when using a torpedo-shaped AUV that operates at constant speed. This situation can lead the planner to generate unfeasible paths that, for instance, require the vehicle to instantaneously change its direction of motion. To cope with this, the planning module has been extended to incorporate such motion constraints [16], as explained below.

Planning paths under motion restrictions are commonly known as kinodynamic motion planning, a term introduced by Donald *et al.* [17], which refers to planning collision-free motions by considering the limits of feasible system maneuvers expressed by differential equations. In the particular case of an AUV, this corresponds to a second-order differential equation of an underactuated system. However, for online path-planning

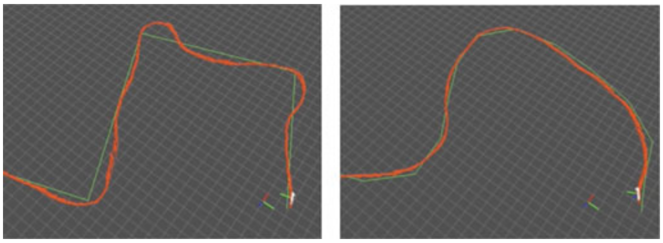


Fig. 10. Real trajectories of Sparus II AUV (in red) following reference trajectories (in green) with a classical line-of-sight controller. Left: reference trajectory does not consider kinematics constraints and the AUV has big overshoots. Right: reference trajectory considers kinodynamic constraints and the AUV is able to follow them with much smaller errors.

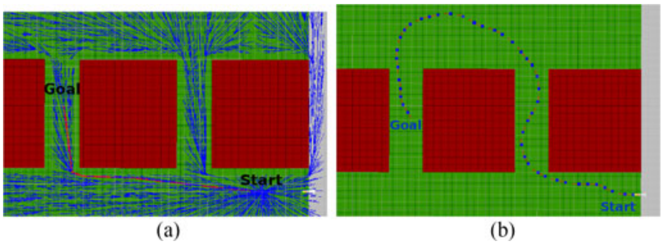


Fig. 11. (a) RRT* expansion without using differential constraints, solution and intermediate states, for a start-to-goal query. (b) RRT solution for a start-to-goal query. The path was obtained by expanding an RRT using (1).

purposes, we represent the torpedo-shaped AUV that navigates at a constant speed and depth as a simple unicycle-like vehicle with kinematics defined by

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix} v.\cos(\psi)\\ v.\sin(\psi)\\ r\end{bmatrix} \qquad (1)$$

where $q = [x, y, \psi]^T$ is the state of the vehicle that includes its 2-D position and orientation with respect to an inertial reference frame, and $\dot{q} = \left[\dot{x}, \dot{y}, \dot{\psi}\right]^T$ is the first time derivative that depends on the state itself and the control inputs, linear/surge speed v, and rate of turn r. From (1), it can be concluded that the configuration space of a torpedo-shaped AUV performing tasks in a plane is 3-D, i.e., each $q \in SE(2) = \mathbb{R}^2 \times \mathcal{S}$. Fig. 10 shows real trajectories of Sparus II AUV in which the necessity of kinodymanic constraints can be shown.

The RRT has proved to be efficient for solving kinodynamic motion planning problems, as the one previously stated for a torpedo-shaped AUV. In the case of expanding a tree of collision-free motions subjected to differential constraints, new states (tree nodes) are obtained by integrating differential equations such as (1), thus generating more feasible (doable) solution paths (see Fig. 11).

Different variants of this approach have been proposed for aerial and terrestrial vehicles to generate smooth and feasible paths. An example of them is one in which a standard RRT is used to find a series of collision-free waypoints, which are then interpolated by using a cubic Bézier spiral to generate a smooth path to be followed by an unmanned aerial vehicle [18]. Another alternative uses an RRT that is expanded by considering not only the vehicle dynamic model, but also the controller behavior [19]. Nonetheless, their major drawback, also observed in Fig. 11(a), is the lack of optimality in any possible metric.

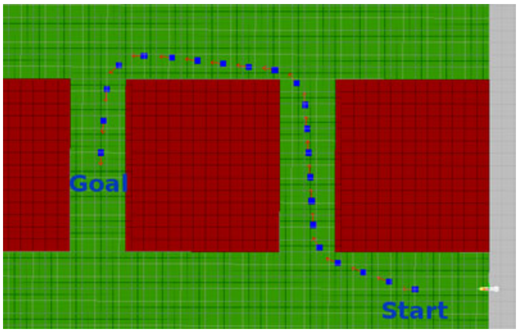


Fig. 12. Solution path for a start-to-goal query using an RRT* and Dubins maneuvers as steering function.

Unlike what occurs with the solution paths calculated by the RRT, the RRT* includes the asymptotic optimality property that checks if reconnecting new state's nearest nodes (configurations) improves their associated cost, thus implying that the probability of obtaining an optimal path increases over time [14]. For doing so, the RRT* requires a steering function which permits such states (nodes, configurations) reconnection, which in the case of systems under differential constraints implies calculating the required input to dynamically evolve the system from a given state to a desired one. However, defining such function may become an intractable nonlinear control problem, especially when considering online computation constraints.

An alternative approach to define such a steering function, we adopted the Dubins vehicle model [20], whose dynamics has the general form presented in (1). Using three possible maneuvers as input, left, straight or right, Dubins curves define six possible paths that characterize the optimal trajectory between two states for a Dubins vehicle. This approach permits us to include motion constraints and use the RRT*. With this approach, the path-planning pipeline is now capable of calculating not only more feasible (doable) motions, but also near optimal ones (see Fig. 12).

C. *Mission Handler*

The third and last functional module that constitutes the path-planning pipeline is the mission handler. This module is in charge of controlling and coordinating the previously explained modules (mapping and planning). It also verifies whether the AUV is prepared to start solving and conducting a task; to do so, this module communicates with other functional modules on the vehicle to verify both that navigation data are correctly being generated and that the vehicle's low-level controllers are not conducting any safety maneuver. After completing the checking stage, the mission handler starts requesting waypoints from the planning module, which, after being received, are adapted and sent to the vehicle's low-level controllers. Finally, this module is also responsible for cancelling any ongoing waypoint if it is notified by the planning module, in case that an imminent collision has been detected.

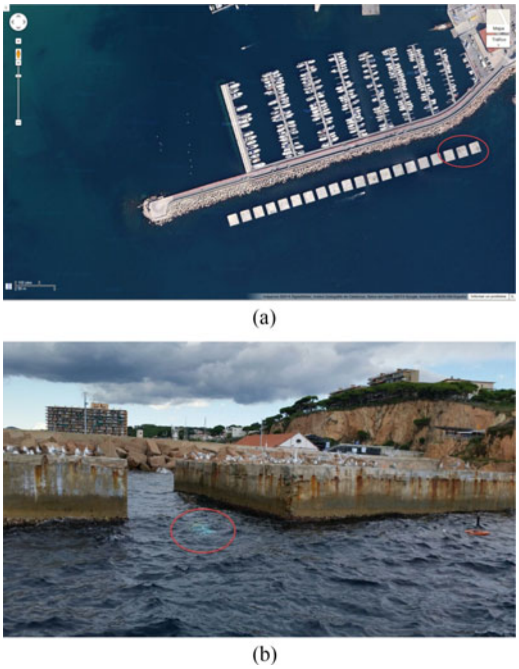


Fig. 13. Structured test scenario. (a) Harbor of Sant Feliu de Guíxols in Catalonia, Spain, where a breakwater structure composed of concrete blocks is demarcated. (b) Sparus II AUV submerged and conducting an autonomous mission.

D. Results

To evaluate both the path-planning and the VP pipelines, we first used the underwater simulator (UWSim) with models of our target environments to do tests of our planning approach before conducting real-world in-water trials. For implementing both pipelines, we also made use of the open motion planning library that offers a convenient framework that can be adapted and extended to specific planning problems [21]. Tests scenarios included a structured and real-world scenario, as well as a virtual and natural-like (nonstructured) environment. Different missions were established to extensively evaluate the effectiveness of our approach. Sections III-D1 and III-D2 present the most relevant results in both types of environments.

1) Structured Test Scenario (A Breakwater Structure): As an example of a structured and challenging scenario, we used the harbor of Sant Feliu de Guíxols. Experiments were conducted in the external and open area of the harbor, in a breakwater structure [marked with a red ellipse as per Fig. 13(a)] that is composed of a series of concrete blocks of 14.5-m length and 12-m width, separated by a 4-m gap with an average depth of 7 m. A virtual version of this scenario was also available for simulation tests before conducting in-water trials.

To prove the effectiveness of the path-planning pipeline in a structured scenario, the Sparus II AUV had to traverse the breakwater structure by solving multiple and successive start-to-goal queries that required the vehicle to navigate between the concrete blocks without any previous knowledge of their location. All queries were defined to conduct missions with a constant depth, since the profiling sonar was located to cover the horizontal plane, thus the motion was restricted to 2-D tasks. Real-world results can be observed in Fig. 14. Further details and information about these experiments can be found in [16].

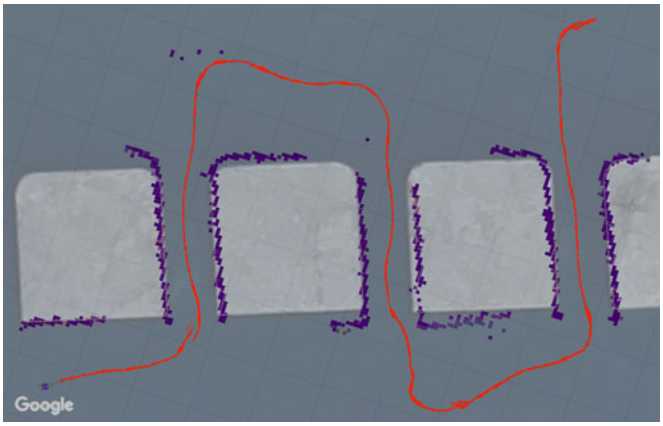


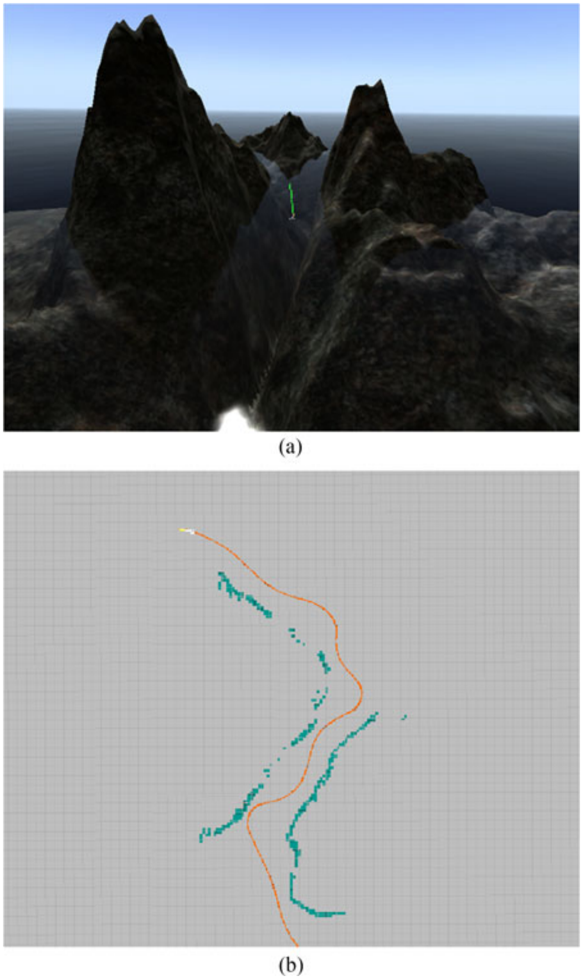
Fig. 14. Examples of missions where Sparus II AUV used the motion planning pipeline to navigate through unexplored environments. The vehicle traversed multiple times a breakwater structure that is composed of a series of concrete blocks.

2) Nonstructured Test Scenario (Underwater Canyon): To evaluate our approach in a nonstructured and challenging scenario, we selected a second virtual scenario that resembles a natural environment. This scenario is composed of rocky formations that create an underwater canyon [see Fig. 15(a)]. As occurred with the breakwater structure scenario, the profiling sonar of the AUV only covered the horizontal plane, which restricted the safe motion of the vehicle to planes of constant depth. For this reason, the navigation was set at a constant depth for the start-to-goal query. Simulation results can be observed in Fig. 15, where the vehicle not only created a map of the surroundings, but also succeeded traversing the canyon.

IV. Online View Planning

The path-planning pipeline explained in Section III constitutes the base for a more specialized pipeline that allows an AUV to autonomously explore and inspect unknown underwater structures. View planning is independent from path planning and, from a higher level, generates the best way points to explore and map the unknown underwater seabed. To provide autonomous underwater exploration capabilities to AUV, some algorithms have been developed by many authors. Most of them rely on prior knowledge about the environment, usually in the form of a rough map, which is used to plan the inspection path (refer to [4], [22]–[25]). From our experience, the need of prior knowledge constitutes a limitation because it restricts the applicability of those methods. Aiming to overcome this limitation, other authors have proposed algorithms that require less prior information about the environment (refer to [26] and [27]). Our work continues this research line.

Our pipeline incorporates an algorithm that has been inspired by VP methods, which aim to determine a suitable set of viewpoints and associated imaging parameters for a specified object reconstruction or inspection task. Given that no previous map of the surroundings is provided, the algorithm iteratively incorporates the sensor data and plans the next best view to fully


(a)

(b)

Fig. 15. Solving start-to-goal query in a virtual scenario of sea rocks. (a) Sea rocks scenario. (b) Conducted trajectory and map perceived during the movement.

and incrementally map an underwater structure or scene. The pipeline inputs are the size, location, and resolution of a bounding box that encloses the structure to be inspected, and also the desired depth of exploration.

The process starts by first incorporating the sensor data into a 2-D grid map, where voxels are labeled according to their possible states (i.e., empty, occupied, viewed, occluded, unexplored, and occplane) as shown in Fig. 16.

1) Empty: a voxel that has been inside the sonar field of view (FOV) with insufficient occupied detections.
2) Occupied: a voxel that has been inside the sonar FOV with sufficient occupied detections.
3) Viewed: an occupied voxel also viewed by the camera.
4) Occluded: a voxel in the sonar FOV but not seen because it is behind an occupied voxel.
5) Unexplored: a voxel that has not been observed yet by the sonar.
6) Occplane: an occluded voxel that is adjacent to an empty voxel.

The processing continues by generating viewpoints according to the map in locations where taking sonar scans or acquiring new images would provide useful information to continue the exploration. The best viewpoint is selected at each iteration by evaluating a metric that takes into account the distance that the robot has to navigate to reach the viewpoint. The closest viewpoint according to this metric is selected.

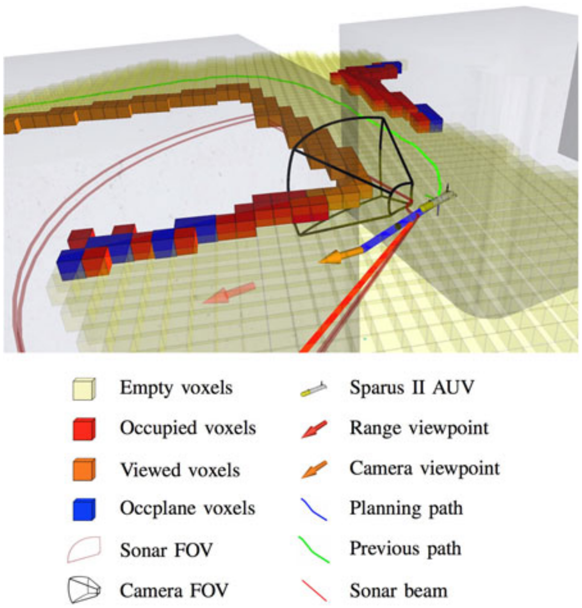


Fig. 16. Example showing all possible labels in a simulated inspection. The FOVs of both sensors are also shown.

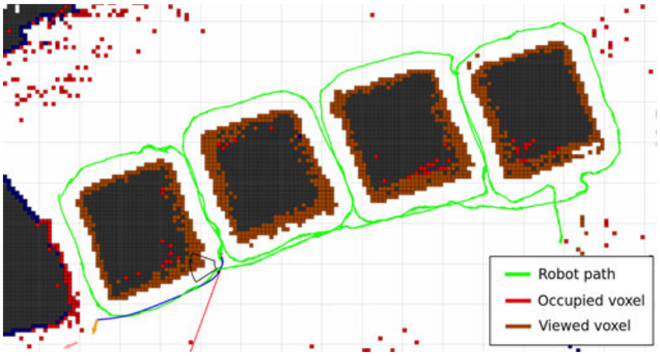


Fig. 17. Real survey in the blocks environment. Robot trajectory is represented in green color. Camera is pointing to the right. First block to be inspected appears on the right of the image.

Finally, collision-free and feasible motions (as those provided by the aforementioned pipeline) are generated to guide the vehicle toward the next best viewpoint.

A. Results

Simulated and real-world trials used a scanning profiling sonar, which is used to build an occupancy map of the surroundings, and an optical camera, which acquires images of the scene. Fig. 17 depicts the results of a real survey performed in the breakwater blocks area. The robot autonomously inspected four concrete blocks (first inspected block appears on the right of the image) in a spiral motion without relying on prior information. Camera was pointing to the right. The experiment

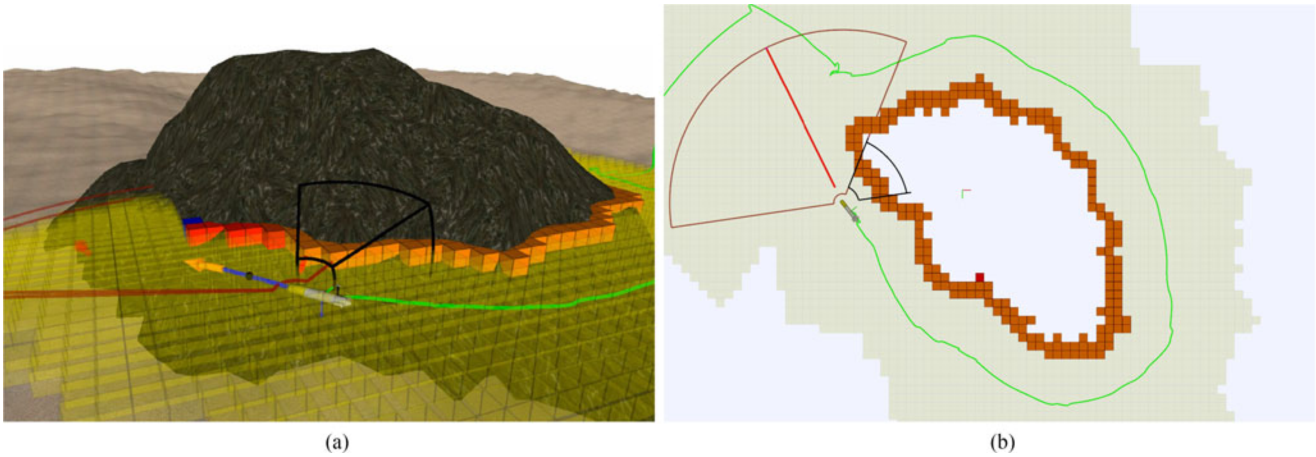


Fig. 18. Simulated survey showing the exploration of a seamount next to St. Feliu de Guíxols. The robot is able to create a 2-D map and extract images of the seamount. At the top of the image, 3-D view of the scene while the inspection is being performed. At the bottom, complete inspection trajectory.

was performed at a constant depth of 1.5 m. The green path in the image represents the robot trajectory. In the map, most of the occupied voxels were labeled as viewed (orange color), which means that images were acquired to apparently cover the complete underwater structure. However, only by processing the actual camera images, the full coverage of the structure can be guaranteed. Real-time image processing for coverage detection is not already available, and it is an interesting feature to be added in the near future.

Fig. 18 shows a simulated exploration of a seamount using the Sparus II AUV. A real bathymetry was used to create the 3-D shape of the environment used in simulation. The seamount is located near the coast of St. Feliu de Guíxols. This isolated underwater boulder is approximately 12-m high, rising from 40 m depth, and its base spans an area of 30 × 15 m.

V. Conclusion

This paper has proposed the platform Sparus II AUV together with a set of path-planning and VP algorithms, as a preliminary solution to inspection AUVs. Online mapping with acoustic profilers is used by path-planning algorithms for generating collision-free trajectories, taking into account the kinematics constraints of the vehicle. VP algorithms are also used to detect 3-D structures and to generate trajectories for visually mapping its boundaries. These new capabilities are performed on-board of Sparus II AUV, a hovering capable AUV able to manoeuvre and work in cluttered environments. Real 2-D results in an artificial harbor structure and simulated natural rocky canyon demonstrated the feasibility of the approach for avoiding or inspecting the underwater environment.

Results have shown the capability of incrementally mapping the unexplored environments while planning paths at the same time, either to solve start-to-goal queries or to fully inspect underwater structures. Working along this line, we plan to extend this approach to 3-D motion by modifying specific parts of each pipelines, avoiding the need of altering the main structure of the framework.

Appendix
Sparus II Hydrodynamic Model

TABLE I
SNAME Notation

DOF	Description	Forces and torques	Linear and angular velocities	Position and Euler angles
1	x axis movement (surge)	X	u	x
2	y axis movement (sway)	Y	v	y
3	z axis movement (heave)	Z	w	z
4	x axis rotation (roll)	K	p	ϕ
5	y axis rotation (pitch)	M	q	θ
6	z axis rotation (yaw)	N	r	ψ

Gravity matrix:

$$G = \begin{pmatrix} (W-B)\sin(\theta) \\ -(W-B)\cos(\theta)\sin(\phi) \\ -(W-B)\cos(\theta)\cos(\phi) \\ By_b\cos(\theta)\cos(\phi) - Bz_b\cos(\theta)\sin(\phi) \\ -Bz_b\sin(\theta) - Bx_b\cos(\theta)\cos(\phi) \\ By_b\sin(\theta) + Bx_b\cos(\theta)\sin(\phi) \end{pmatrix}$$

where

1) $\mathbf{W}$ is the weight;
2) $\mathbf{B}$ is the buoyancy;
3) $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are the pitch and roll angles, respectively;
4) $\mathbf{x_b}$, $\mathbf{y_b}$, and $\mathbf{z_b}$ are the position of the center of buoyancy with respect to the center of gravity.

Lineal damping coefficients:

$$D_L = -\begin{pmatrix} X_u & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r \end{pmatrix}.$$

Quadratic damping coefficients:

$$D_Q = -\begin{pmatrix} X_{uu}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{vv}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{ww}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{pp}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{qq}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{rr}|r| \end{pmatrix}.$$

Damping matrix:

$$D = D_L + D_Q.$$

Rigid body mass and inertia matrix:

$$M_{RB} = \begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{pmatrix}.$$

Added mass coefficients:

$$M_A = -\begin{pmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot{r}} \end{pmatrix}.$$

Mass matrix:

$$M = M_{RB} + M_A.$$

Rigid body mass Coriolis:

$$C_{RB} = \begin{pmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_{zz}r & -I_{yy}q \\ -mw & 0 & mu & -I_{zz}r & 0 & I_{xx}p \\ mv & -mu & 0 & I_{yy}q & -I_{xx}p & 0 \end{pmatrix}.$$

Added mass Coriolis:

$$C_A = \begin{pmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{pmatrix}.$$

Coriolis matrix:

$$C = C_{RB} + C_A.$$

Fins matrix:

$$\tau_{\text{fins}} = \begin{pmatrix} -0.5A_{\text{fins}}K_{\text{fins}}\rho_{\text{wat}}|\sin(\delta)|u|u| \\ 0 \\ -0.5A_{\text{fins}}K_{\text{fins}}\rho_{\text{wat}}\sin(\delta)u|u| \\ \quad -0.5A_{\text{fins}}K_{\text{fins}}\rho_{\text{wat}}\cos(\delta)w|w| \\ 0 \\ 0.5A_{\text{fins}}K_{\text{fins}}\rho_{\text{wat}}\sin(\delta)u|u|x_{\text{fins}} \\ \quad +0.5A_{\text{fins}}K_{\text{fins}}\rho_{\text{wat}}\cos(\delta)w|w|x_{\text{fins}} \\ 0 \end{pmatrix}$$

where

1) $\mathbf{A}_{\text{fins}}$ is the fins area;
2) $\mathbf{K}_{\text{fins}}$ is the fins coefficient;
3) $\boldsymbol{\rho}_{\text{wat}}$ is the water density;
4) $\boldsymbol{\delta}$ is the fins orientation;
5) $\mathbf{x}_{\text{fins}}$ is the distance between the fins and the center of gravity.

Main dynamics equation:

$$M\dot{v} + C_{(v)}v + D_{(v)}v + G = \tau_{\text{thr}} + \tau_{\text{fins}} + \tau_{\text{ext}}.$$

Model parameters: see Tables II–V.

TABLE II
DAMPING PARAMETERS

Parameter	Value	Units	Description
X_u	-9.9239	$\frac{\text{N·s}}{\text{m}}$	Surge linear damping
X_{uu}	-10.1728	$\frac{\text{N·s}^2}{\text{m}^2}$	Surge quadratic damping
Y_v	*	$\frac{\text{N·s}}{\text{m}}$	Sway linear damping
Y_{vv}	*	$\frac{\text{N·s}^2}{\text{m}^2}$	Sway quadratic damping
Z_w	0.0	$\frac{\text{N·s}}{\text{m}}$	Heave linear damping
Z_{ww}	-259.8366	$\frac{\text{N·s}^2}{\text{m}^2}$	Heave quadratic damping
K_p	*	$\frac{\text{N·m·s}}{\text{rad}}$	Roll linear damping
K_{pp}	*	$\frac{\text{Nms}^2}{\text{rad}^2}$	Roll quadratic damping
M_q	*	$\frac{\text{N·m·s}}{\text{rad}}$	Pitch linear damping
M_{qq}	*	$\frac{\text{N·m·s}^2}{\text{rad}^2}$	Pitch quadratic damping
N_r	0.0	$\frac{\text{N·m·s}}{\text{rad}}$	Yaw linear damping
N_{rr}	-53.2837	$\frac{\text{N·m·s}^2}{\text{rad}^2}$	Yaw quadratic damping

TABLE III
MASS AND INERTIA PARAMETERS

Parameter	Value	Units	Description
m	52.0	kg	Mass
I_{xx}	0.38545	kg·m^2	x-axis inertia
I_{yy}	11.28606	kg·m^2	y-axis inertia
I_{zz}	11.28606	kg·m^2	z-axis inertia
$X_{\dot{u}}$	-41.33	kg	Surge added mass
$Y_{\dot{v}}$	*	kg	Sway added mass
$Z_{\dot{w}}$	-68.00	kg	Heave added mass
$K_{\dot{p}}$	*	kg·m^2	Roll added mass
$M_{\dot{q}}$	*	kg·m^2	Pitch added mass
$N_{\dot{r}}$	-101.00	kg·m^2	Yaw added mass

TABLE IV
WEIGHT AND BUOYANCY PARAMETERS

Parameter	Value	Units	Description
W	509.95	N	Weight
B	524.45	N	Buoyancy (according to density)
x_b	0.0	m	Center of buoyancy, x coordinate
y_b	0.0	m	Center of buoyancy, y coordinate
z_b	-0.02	m	Center of buoyancy, z coordinate

TABLE V
FINS PARAMETERS

Parameter	Value	Units	Description
A_{fins}	0.0180	m^2	Fins area
K_{fins}	8.00		Fins coefficient
x_{fins}	-0.51	m	Fins to CoG distance

REFERENCES

[1] M. Carreras *et al.*, "SPARUS II, design of a lightweight hovering AUV," in *Proc. Int. Workshop Mar. Technol.*, 2013, pp. 152–155.

[2] M. Quigley *et al.*, "ROS: An open-source robot operating system," in *Proc. IEEE Int. Conf. Robot. Autom. Workshop Open Source Software*, Kobe, Japan, May 2009, pp. 1–6.

[3] B. Bingham *et al.*, "Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle," *J. Field Robot.*, vol. 27, no. 6, pp. 702–717, Nov. 2010. [Online]. Available: http://doi.wiley.com/10.1002/rob.20350

[4] F. S. Hover *et al.*, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, Nov. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912461059

[5] E. Galceran, R. Campos, N. Palomeras, D. Ribas, M. Carreras, and P. Ridao, "Coverage path planning with real-time replanning and surface reconstruction for inspection of three-dimensional underwater structures using autonomous underwater vehicles," *J. Field Robot.*, vol. 32, no. 7, pp. 952–983, 2014. [Online]. Available: http://dx.doi.org/10.1002/rob.21554

[6] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE J. Ocean. Eng.*, vol. 26, no. 2, pp. 240–251, Apr. 2001. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=922790

[7] T. Maki, H. Mizushima, H. Kondo, T. Ura, T. Sakamaki, and M. Yanagisawa, "Real time path-planning of an AUV based on characteristics of passive acoustic landmarks for visual mapping of shallow vent fields," in *Proc. MTS/IEEE OCEANS*, Vancouver, BC, Canada, 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4449321

[8] M. Carreras, J. D. Hernndez, E. Vidal, N. Palomeras, and P. Ridao, "Online motion planning for underwater inspection," in *Proc. 2016 IEEE/OES Auton. Underwater Veh.*, Nov 2016, pp. 336–341.

[9] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Chichester, U.K.: Wiley, Apr. 2011. [Online]. Available: http://doi.wiley.com/10.1002/9781119994138

[10] N. Palomeras, A. El-Fakdi, M. Carreras, and P. Ridao, "COLA2: A aontrol architecture for AUVs," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 695–716, Oct. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6263248

[11] M. Prats, J. Perez, J. J. Fernandez, and P. J. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2577–2582. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6385788

[12] J. D. Hernández, E. Vidal, G. Vallicrosa, E. Galceran, and M. Carreras, "Online path planning for autonomous underwater vehicles in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, May 2015, pp. 1152–1157. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7139336 http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7139336

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, Feb. 2013. [Online]. Available: http://link.springer.com/10.1007/s10514–012-9321-0

[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011. [Online]. Available: http://ijr.sagepub.com/content/30/7/846.short

[15] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/02783640122067453

[16] J. D. Hernández, M. Moll, E. Vidal, M. Carreras, and L. E. Kavraki, "Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Daejeon, South Korea, 2016, pp. 1313–1320.

[17] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, pp. 1048–1066, Nov. 1993. [Online]. Available: http://portal.acm.org/citation.cfm?doid=174147.174150

[18] K. Yang and S. Sukkarieh, "3D smooth path planning for a UAV in cluttered natural environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2008, pp. 794–800.

[19] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5175292

[20] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957. [Online]. Available: http://www.jstor.org/stable/2372560

[21] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot., Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6377468

[22] E. Galceran, "Coverage path planning for autonomous underwater vehicles," Ph.D. dissertation, Comput. Vision Robot. Inst., Universitat de Girona, Girona, Spain, 2014.

[23] A. Kim and R. M. Eustice, "Next-best-view visual SLAM for bounded-error area coverage," in *Proc. IROS Workshop Active Semant. Perception*, Vilamoura, Portugal, 2012. [Online]. Available: http://robots.engin.umich.edu/publications/akim-2012a.pdf

[24] A. Bircher *et al.*, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proc. Int. Conf. Robot. Autom.*, 2015, pp. 6423–6430.

[25] P. S. Blaer and P. K. Allen, "Data acquisition and view planning for 3-D modeling tasks," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, 2007, pp. 417–422.

[26] D. P. Williams, F. Baralli, M. Micheli, and S. Vasoli, "Adaptive underwater sonar surveys in the presence of strong currents," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 2016, pp. 2604–2611.

[27] J. I. Vasquez-Gomez, E. Lopez-Damian, and L. E. Sucar, "View planning for 3D object reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4015–4020.

Marc Carreras (M'09) received the B.Sc. degree in industrial engineering and the Ph.D. degree in computer engineering from the University of Girona, Girona, Catalonia, Spain, in 1998 and 2003, respectively.

From 1999 until 2016, he has participated in 16 research projects (7 European and 9 National), is author of more than 90 publications, has supervised 5 Ph.D. dissertations, and has participated in several European AUV competitions (5 times winner). He is currently an Associate Professor with the Computer Vision and Robotics Institute, University of Girona. His research activity is mainly focused on underwater robotics in research topics such as intelligent control architectures, robot learning, path planning, and AUVs.

Juan David Hernández (M'05) received the B.Sc. degree in electronic engineering from Pontifical Xavierian University, Bogotá, Colombia, in 2009, the M.Sc. degree in robotics from the Technical University of Madrid, Madrid, Spain, in 2012, and the Ph.D. degree in technology (robotics) from the University of Girona, Girona, Spain, in 2017.

He is currently a Postdoctoral Fellow at the Computer Science Department, Rice University, Houston, TX, USA. His research activity is focused on motion planning algorithms under motion and online computation constraints.

Dr. Hernández is a member of the IEEE Robotics and Automation Society.

Eduard Vidal received the B.Sc. degree in industrial engineering from the University of Girona, Girona, Spain, in 2014 and the M.Sc. Erasmus Mundus Master degree in computer vision and robotics (VIBOT) from the University of Burgundy, Dijon, France, University of Girona, Girona, and Heriot-Watt University, Edinburgh, U.K. He is currently working toward the Ph.D. degree at the Underwater Robotics Research Center, part of the Computer Vision and Robotics Institute, University of Girona.

His research interests include motion and view planning for autonomous vehicles.

Narcís Palomeras (M'17) received the B.S. degree in computer science and the Ph.D. degree in computer engineering from the University of Girona, Girona, Spain, in 2004 and 2011, respectively.

He is currently a Postdoctoral Fellow at the Department of Computer Engineering, University of Girona, Girona, Spain, and a member of the Underwater Robotics Laboratory in the Computer Vision and Robotics Group. He has participated in several research projects (both national and European) related with underwater robotics and has taken part in several European AUV competitions. His research activity is mainly focused on underwater robotics in research topics such as intelligent control architectures and mission control.

David Ribas received the M.Sc. and Ph.D. degrees, both in industrial engineering, from the University of Girona, Girona, Spain, in 2003 and 2008, respectively.

He is currently a Researcher at the Department of Computer Engineering, University of Girona, and a Member of the Research Center at Underwater Robotics. He is involved in national and European projects about underwater robotics and some technology transference projects about realtime and embedded systems. His research interests include the development of AUVs, and more particularly to the autonomous navigation problem using simultaneous localization and mapping techniques.

Pere Ridao (M'04) received the M.Sc. degree in computer science from the Technical University of Catalonia, Barcelona, Spain, in 1993 and the Ph.D. degree in computer engineering from the University of Girona, Girona, Spain, in 2001.

He is currently an Associate Professor at the Department of Computer Engineering, University of Girona, and the Head of the Research Center in Underwater Robotics, Girona. His research interest focusses on underwater robotics in research topics, such as intelligent control architectures, UUV modeling and identification, simulation, navigation, mission control, and real-time systems.

Dr. Ridao is a Member of the IFAC's Technical Committee on Marine Systems, a Member of the Editorial Board of Springer's *Intelligent Service Robotics Journal*, a Secretary of the Spanish OES chapter, and also a Board Member of the Spanish RAS chapter.