Ocean Engineering 269 (2023) 113514

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A simulator of underwater glider missions for path planning[☆]

Aurélien Merci [a,b,c,*], Cédric Anthierens [b], Nadège Thirion-Moreau [c], Yann Le Page [a]

[a] Alseamar, 60 Av. Olivier Perroy, Rousset, 13790, France
[b] Université de Toulon, COSMER, Av. de l'Université, La Garde, 83130, France
[c] Université de Toulon, AMU, CNRS, LIS Toulon, UMR 7020, France

ARTICLE INFO

Keywords:
Underwater gliders
Mission simulator
Path planning
Environment modeling
Vehicle routing problems

ABSTRACT

In this article, we present a mission simulator developed for the Alseamar SeaExplorer underwater gliders. By taking into consideration a 4D time-varying environment, it provides an estimation of the most important metrics: the battery level, the mission duration and the distance traveled by the glider.

The main strengths of this simulator are first its use upstream as a tool to aid the glider pilots to define a feasible mission plan. Secondly, through its interoperability with the Alseamar mission management system "GLIMPSE", it works as an internal model (replanning) during the mission. Finally, it generates adjacency matrices of weighted graphs on which our high level path planning algorithms for a single glider as well as a fleet are based.

This simulator is compared to a real experimental mission in order to confirm its accuracy and efficiency.

1. Introduction

Autonomous Underwater Gliders (AUGs) were invented in order to address the growing need for measures performed in marine environments on large spatial and temporal scales. They were designed to be easy to use, scalable and relatively "cheap" compared to other underwater vehicles. The concepts of underwater gliders were defined by H. Stommel in 1989 (Stommel, 1989). Their original goal was to collect water column data profiles with wide spatio-temporal coverage (thousands of kilometers and weeks to months of endurance). Thanks to buoyancy changes, they glide up and down, alternating between ascent and descent cycles. Technological breakthroughs in low-power electronics, batteries and sensors with extended capacities of satellite geolocation have paved the way to the development of the first four commercial underwater gliders: "Slocum" (Webb et al., 2001), "Spray" (Sherman et al., 2001), "Seaglider" (Eriksen et al., 2001) and "SeaExplorer" (Fommervault et al., 2018). In the balance of this article, we will focus on the latter, developed by Alseamar.

Nowadays, the gliders are classified under the category of Autonomous Underwater Vehicle (AUV). A glider is profiled for hydrodynamics in order to perform long endurance missions. According to Meyer (2016), the nose cone embeds the payload whereas the rear part is dedicated to the navigation control. Finally, the tail is a foldable mast that allows the glider to communicate its position and data to/from a terrestrial basis when it is surfacing (around every three hours).

The SeaExplorer can be customized with a large variety of sensors. The aim is to respond to the different marine applications that can be targeted: observation of coastal, meso- and submesoscale dynamics, mixing processes and transport of water and energy, impact of glider data assimilation on ocean models, acoustic detection of biological and geological activity, sediment transport/resuspension (see Meyer (2016) for an overview). The large scale of these oceanographic phenomenons also explains the growing interest in the use and the coordination of fleets of underwater gliders (Lekien et al., 2008; Leonard et al., 2010; Alvarez and Mourre, 2014; Barbier et al., 2019). The main objective of using a glider fleet is to provide the best possible and up-to-date view of an operating zone and to provide data that can be assimilated by ocean models and thus improve forecasts. The glider trajectories can be predetermined but can also be spatially refined or responsive either to sensor measurements or to marine environmental changes. Thus, it is useful to test these trajectories in simulation to assess the relevance of a mission plan for a fleet of gliders during the planning phase but also during the mission itself for the supervising.

Gliders operate at relatively low speed (around a half knot) which makes them sensitive to water currents. It is the reason why it is crucial to develop a glider mission simulator that can take into consideration a 4D environmental model (3D in space in addition to 1D in time).

☆ This work has been carried out in the framework of a CIFRE-DGA grant (n°2020858) allowed by the French Ministry of the Armed Forces and the Defense Innovation Agency.

* Corresponding author at: Alseamar, 60 Av. Olivier Perroy, Rousset, 13790, France.
E-mail address: amerci@alseamar-alcen.com (A. Merci).

https://doi.org/10.1016/j.oceaneng.2022.113514
Received 31 August 2022; Received in revised form 1 December 2022; Accepted 21 December 2022
Available online 29 December 2022
0029-8018/

Most recent research works on the underwater gliders focus on the development of path planning algorithms taking into consideration a time-varying environment (Lan et al., 2022; Cai et al., 2021; Ji et al., 2022; Liu et al., 2022; Lan et al., 2021) and on the formation control and the coordination of multiple underwater gliders (Wen et al., 2022; Ma et al., 2022). In our case, the longer-term objective is to make the operating system of the gliders more autonomous.

The existing state-of-the-art simulators dedicated to robotics can be classified into three main categories, depending on the type of environment that is considered: mobile ground, aerial and marine. Most of them address both mono and multi-vehicle problems. They also emulate different types of data acquisition from many sensors and can be coded in different languages (the most popular ones being C++, Python and Java).

They are introduced by complexity order. First, in mobile ground robotics, the environment is largely known and can be imported from digital elevation models, SDF meshes, OpenStreetMap. The most famous simulators are Gazebo, CoppeliaSim (Rath et al., 2018; Tai et al., 2017; Chen et al., 2017), Webots and CARLA (Codevilla et al., 2017; Zhang et al., 2018; Dosovitskiy et al., 2017). Notice that Webots and Gazebo are used both for ground (Winkler et al., 2018; Bellicoso et al., 2018-07; Takaya et al., 2016; Zhao et al., 2015; Juang and Yeh, 2018) and aerial robotics (Koenig and Howard, 2004; Schmittle et al., 2018; Imanberdiyev et al., 2016; Mahdoui et al., 2019; Michel, 2004). Two other simulators are specifically dedicated to aerial robotics: AirSim (Shah et al., 2017) and Flightmare (Song et al., 2020). They allow to run the simulation of high-fidelity environments (warehouses, forests, etc.). Finally, concerning the marine environment, it remains the most complex one because it focuses on a relatively unknown and highly fluctuating environment. Three simulators can be mentioned: UUV Simulator (Manhães et al., 2016), UWSim (Prats et al., 2012), StoneFish (Cieślak, 2019). These simulators compute the full dynamics and hydrodynamics coefficients of autonomous underwater vehicles. The SeaExplorer model has not been implemented in these simulators so far. Thrusters, sensors (such as DVL, pressure sensors, USBL, sonars, acoustic communication device, cameras and so on) and robot arm can be simulated. They also simulate complex and realistic underwater environments such as currents, waves, seabed, lakes, etc. Until now, they are not dedicated to path planning and vehicle routing and do not provide the important metrics to be considered in such problems (mission duration, remaining energy and travel distance).

The main objective of this article is to introduce a new full mission simulator for SeaExplorers gliders in a time-varying large scale environment. This is an ongoing problem as highlighted by recent publications such as Phoemsapthawee et al. (2013) and Grande et al. (2021). There is also a real need for this kind of simulator in order to help the SeaExplorer's pilots to prepare the mission (offline mode) but also during the execution of the mission (online mode). Indeed the gliders may operate over several weeks/months through different types of water current or different water densities which affect the course of the mission.

The simulator that we developed takes into account:

- Environmental data from the Marine Copernicus database (Marine Copernicus database, 2015),
- The flight profiles of the SeaExplorer gliders,
- A mono and multi-vehicles configuration depending on the kind of undertaken mission,
- The output of high level path planning algorithms (i.e. Hamiltonian path Rahman and Kaykobad, 2005 for example, etc.)

and provides:

- Information and forecast on consumption using models,
- The interoperability with the Alseamar mission management system GLIMPSE (GLIder Mission Piloting SystEm Besson et al., 2019). This means a precious decision aid for the end-users i.e. the SeaExplorer's pilots during both the mission preparation (path planning and choice of a feasible mission plan) and the progress of the mission (internal model and replanning),
- A metrics report through a human–machine interface,
- When weighted graph (Bondy and Murty, 1976) based path planning algorithms are considered, it is possible to use the simulator to build the adjacency matrix (measure of the cost between any points and their neighbors taking into account either the distance or the travel time or the energy consumption),
- A comparison between the different mission plans provided by high level path planning algorithms but also between the different scenarios depending on the navigation profiles and the embedded sensors.

It is very interesting to have such a tool to reduce the cost and the time dedicated to the mission preparation phase. The simulator also needs to be scalable in order to easily integrate simulated data acquisition for real time adaptive behavior.

The article is organized as follows. After an introduction, the second section is dedicated to the presentation of the Alseamar SeaExplorer gliders and their current operating process. In the third section, the simulator requirements are presented. Then, in the fourth section, the framework and the different levels of the simulator are detailed. In the fifth section, computer simulations are performed to illustrate the behavior and relevance of our simulator. An experimental validation is also carried out through the comparison with a real experimental mission. In the last section, an evaluation of the outputs of a path planning algorithm through the simulator is presented. Finally, a conclusion is drawn and perspectives are delineated on the multi-vehicles path planning generation.

2. SeaExplorer: principle and current operating process

2.1. The vehicle model

2.1.1. General presentation

The Alseamar SeaExplorer glider is composed of a payload section, a vehicle section and a communication section as depicted in Fig. 1 (for more details on its technical specifications see SeaExplorer specifications (2014) and Table 1).

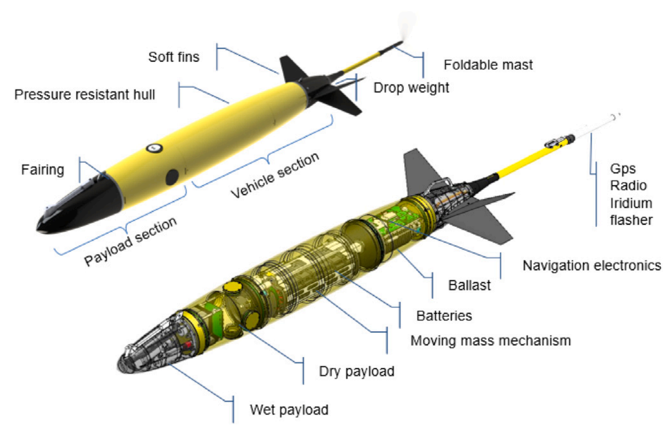


Fig. 1. Open glider.

For the sake of battery saving and thus for the autonomy, the SeaExplorer does not rely on an energy consuming means of propulsion. The embedded AHRS (Attitude Heading Roll Sensor) provides measures of the roll angle, the pitch angle and the heading whereas the depth is separately measured by a depth sensor. So Fig. 2 gives a glimpse of the way the glider works with regards to its control level.

Underwater displacement is done by "hovering" between the surface and the desired depth, using gravity to descend along the water column and Archimede's force to rise. Indeed, the vehicle has a bladder

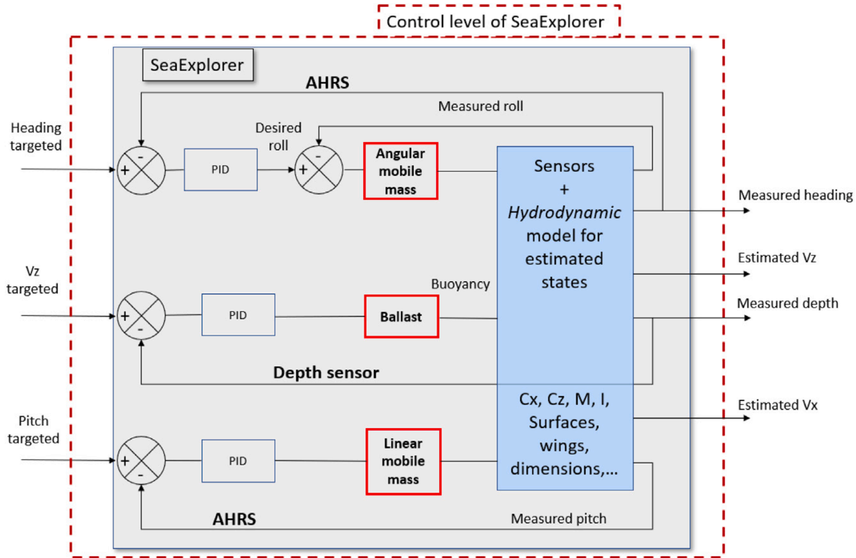


Fig. 2. Control level of SeaExplorer.

allowing a variation of its volume, and therefore its density. For the ascending phase, the bladder is inflated thanks to a motorpump which is controlled to generate a constant flow value independently of the surrounding pressure due to the depth of the vehicle (nevertheless the energy consumption of this actuator varies with this pressure thus the depth). The duration of inflation is modulated to reach the desired value of buoyancy.

As depicted on Fig. 2, all these three actuators are controlled thanks to their own PID controller. During a turn (angular moving mass) or a flight (ballast and pitch due to linear moving mass) the changes of buoyancy, heading and pitch imply the three actuators to be driven by an anti-windup PID controller until the error has significantly decreased. Because these actuators are not reversible (the worm screw makes it irreversible), such a method is interesting to limit the energy consumption. It also means that the time during which the actuators are on is finely calculated in order to provide the glider with the desired vertical speed, the desired heading and the desired pitch. The sampling time of the actuators loop is 20 s which is low enough in comparison to the vehicle dynamics (a half turn lasts about 330 s whereas a transition between ascending and descending phases takes about 90 s). The linear moving mass (related to pitch) and the angular moving mass (related to roll) are both controlled by a PID controller which are supplied by sensors signals feedback i.e. the pitch and the heading.The glider dives or rises with a target pitch angle denoted by α (see Fig. 3) during its "yos" (SeaExplorer cycles). Thus it converts its vertical speed (V_z in Fig. 3) into horizontal speed (V_x in Fig. 3).

These cycles are composed of descent and ascent navigation phases as described in Fig. 4, between a minimal immersion (z_t) and a maximal immersion (z_b) given a "multi dives" parameter (sr). When the glider reaches the surface immersion (z_s), the moving mass and the ballast are set to deploy the antenna out of the water in order to start the communicating phase (state 116 "Transmitting" - Fig. 4). The GPS position is computed and the exact location is transmitted by radio frequency or Iridium to the base. Navigation orders and data are received and transmitted.

A summary of the main specifications of the SeaExplorer glider can be found in the Table 1.

2.1.2. Navigation phases

Over one "yo", the glider follows different navigation sequences, and thus different phases numbered from 100 to 118 (cf. Fig. 4).

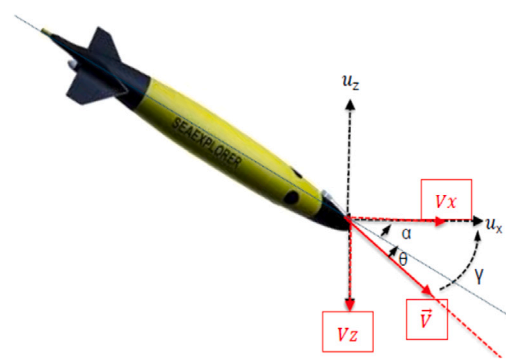


Fig. 3. Velocities and angles description.

Table 1
SeaExplorer specifications.

Body size (DxL)	0.25 m × 2 m + 1 m foldable antenna
Wingspan	56.5 cm Wingless design
Weight	59 kg in air
Depth rating	1000 m
Ballast volume	1000 cc (±500 ml)
Speed	Nominal 0.5 kt/Maximum 1 kt
Battery	Primary Lithium
Typical range (Endurance)	1300 km (64 days)/3200 km (160 days) with a GPCTD-DO sensor sampling at 4 s

- State 115 ("Surfacing"): the moving mass and the ballast are set to deploy the antenna out of the water for communicating. The vehicle inflates the bladder (+400 ml position – up to 500 ml) and the moving mass moves forward to pitch down.
- State 116 ("Transmitting"): the glider is communicating at the surface. The GPS position is computed and the exact location is transmitted by radio frequency or Iridium. Navigation orders and data can be received and transmitted.
- State 110 ("Inflecting down"): the glider starts inflecting down. The ballast is adjusted (oil bladder deflated) to make the glider heavy and the moving mass moves forward to make it pitch down.
- State 100 ("Going down"): the ballast is in its diving position. Depending on the configuration, it adjusts the pitch angle and the heading.

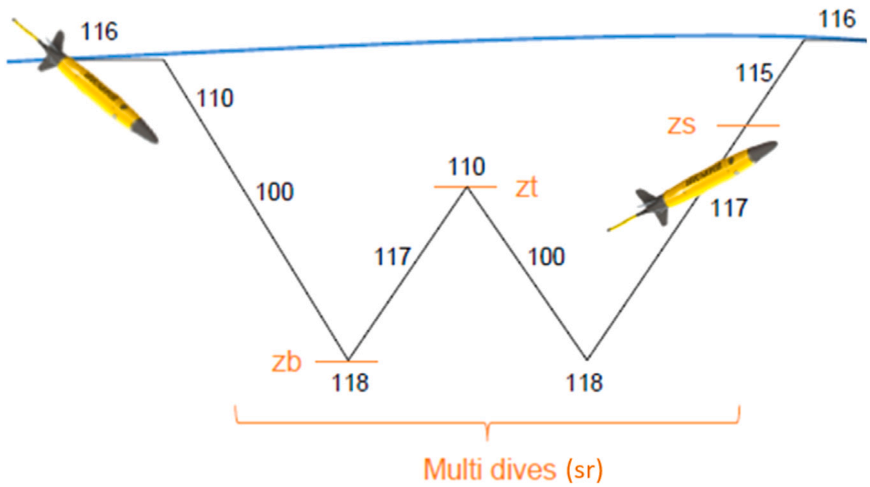


Fig. 4. Navigation phases.

- State 118 ("Inflecting up"): once depth or altitude reached, the glider starts inflecting up. The ballast is adjusted (oil bladder inflated) to make the glider buoyant and the moving mass moves backward to make it pitch up.
- State 117 ("Going up"): the ballast is kept in the same position while returning to the surface. The glider controls the pitch angle and the heading.

2.1.3. Hydrodynamic and energy consumption

The role of the hydrodynamic model is to compute the angle of attack (θ) which varies with the pitch (α). Such a model depends on the characteristics of the vehicle (dimensions, weight, positions of the centers of buoyancy and mass, 3D drag coefficient ...). Most of the time, the glider moves with a constant speed (between two changes of direction), this allows assuming it in a static motion. So by considering the drag forces D_x, D_y, the weight P, the buoyancy B and the fins lift L, the model gives the pitch, the glide speed and the glide angle (Fig. 5). Because of the non-linearity of the drag forces and the speeds, the hydrodynamic model of the SeaExplorer relies on specific tables to evaluate all its current characteristics. This accurate model provides the vertical and horizontal speeds (respectively V_z, V_x) used to estimate the underwater position of the glider during a dive.

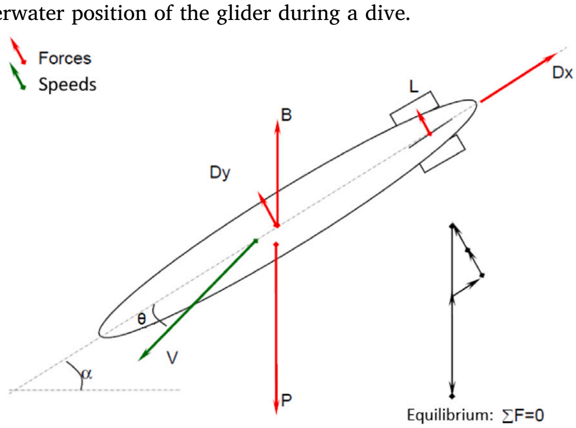


Fig. 5. Static resolution.

The intrinsic mechanical parameters of the vehicle are included in the hydrodynamic model. The outputs of this block are related to the glider displacement either estimated or measured. During the going up and down phases, the energy consumption is essentially due to the payload. The rejection of potential disturbances ensured by the heading and the pitch controllers is negligible. Moreover since the hydraulic circuit of the ballast is equipped with a motorpump and a valve, and it features neither leakage nor fluid compressibility, its consumption is null during these navigation phases except during the transition ones (inflecting up and down, transmitting and surfacing). Such models considered in the simulator are based on experimental data measured during several missions. It means that during the going up and down phases, the consumption is assumed to be the addition of the consumption of all the components that have been turned on.

The actuators, moving masses and ballast, are the main reasons of the energy consumption. As for the moving masses, the PID controller lead them to maintain a extremal position while the glider is changing its orientation (pitch and/or heading) until the error of orientation becomes low. It means that independently to the range of the orientation change, the stroke of the moving masses is always the same, thus the energy consumption too according to the considered phase. It does not depend on the depth contrarily to the ballast actuation whose energy consumption straightly varies linearly with the water pressure, thus the depth. The power considered is related to the depth then the energy is the product of the power and the time of actuation.

To introduce the simulator, the whole glider SeaExplorer is represented by a physical model (hydrodynamics) and experimental behavioral model and control model (for each control variable namely the buoyancy, the roll and the pitch). Such a modeling is relevant for the simulator we aim at designing because its goals focus on the mission preparation, the path planning and navigation rather than on how to improve the hydrodynamics model or the accuracy of the control of the actuators.

2.1.4. Water current compensation

The glider continuously controls its heading thanks to an improved PID controller feeded by the compass data, which drives the roll angle of the vehicle. Nevertheless, the eventual water currents may generate a drift on the glider. In order to limit the shift of trajectory, an estimation of the average current is computed from the last "yo" the glider experienced. So the actual desired heading integrates the estimated water current to compensate the potential drift. This current compensation is updated at each surfacing.

2.2. Control-command current process

Before starting a glider mission, a vehicle configuration has to be defined. It means choosing the type of batteries and a set of sensors required to properly run the mission. These will constitute the payload implemented on the vehicle. Presently, the planning and navigation tasks are done by a pilot all along the mission to define the steering of the gliders through the mission management system named GLIMPSE (snapshot provided in Fig. 6).

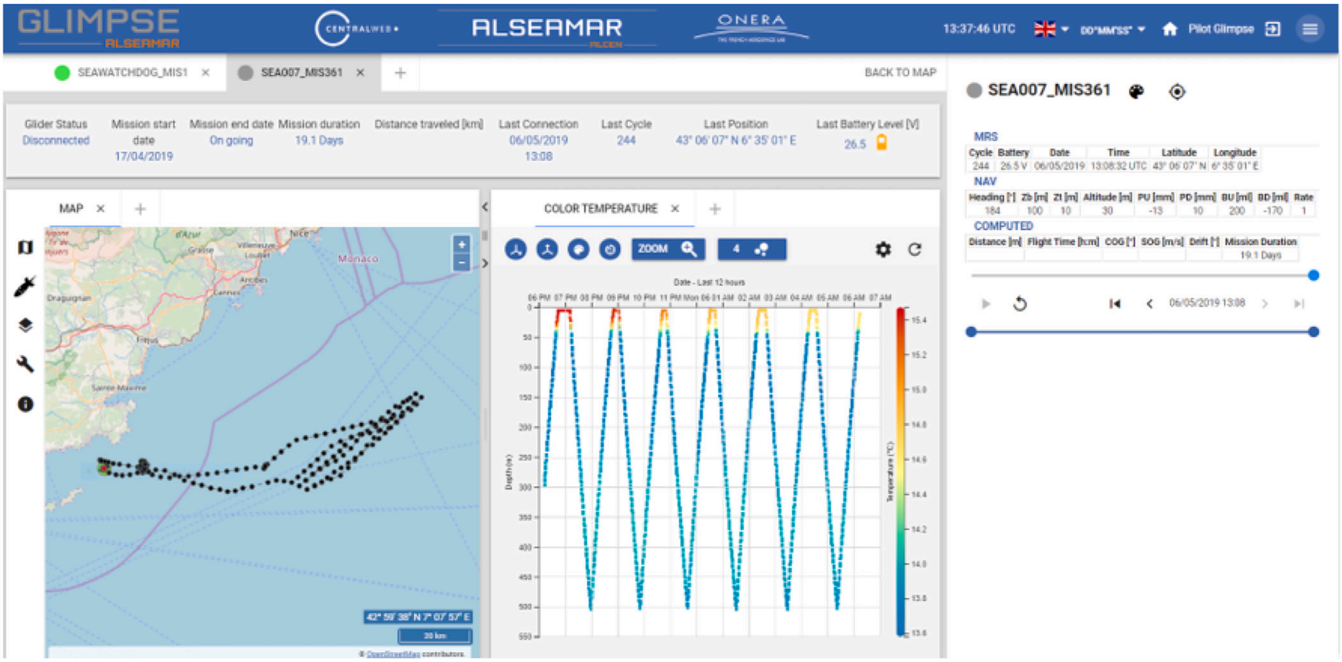


Fig. 6. Mission management system.

Such a tool transmits the commands from the pilots to the gliders and moreover it collects and displays all the states of the vehicle for the pilots. This communication phase is run every surfacing and it lasts about 5 min. All monitored data are therefore frequently updated and, so, help the pilot to refine the mission parameters if necessary. These data are accessible during the mission in order to get a synoptic view of what is actually occurring. The mission plan can be adapted according to what is perceived by the vehicles.

A SeaExplorer's pilot can define a position to target by the glider, called a "waypoint", which is defined by its latitude and longitude coordinates and by a validation distance. If the glider enters this validation area, the waypoint is regarded as hit. When a waypoint is defined, the mission management system recalculates and sends the new navigation heading for the vehicle at each surfacing in order to hit the waypoint with a predefined navigation profile for the current segment. A segment is a path from an initial waypoint to a target waypoint. The mission management system GLIMPSE lets the pilot define a "single waypoint" (manual mode) or "mission plan" mode (multi-waypoints path) defined by a list of waypoints and a predefined flight profile.

Thanks to its internal models and the measurements from the navigation sensors, the glider estimates its location every 20 s. The real position when surfacing and the trajectories are also displayed on the mission management system during the mission. They are displayed using geographic coordinates since the shape of the Earth can have an influence on the precision of the trajectories.

Either for the mission preparation phase or during the supervision of the missions, the Alseamar's pilots have access to a variety of maps such as water current maps, bathymetric maps, temperature maps. These maps are available from the Marine Copernicus database and help the pilots to define an achievable mission plan for the gliders. They also respond to a real operational need. These products can be obtained through experimental measurements led on an area or thanks to a numerical model (e.g. water current model). The latter gives a larger temporal coverage than the products resulting from observations. It is also possible to retrieve the forecasts for the next few days and to have the measurements available in depth on a certain part of the water column. Thus, the current models remain the most used products by Alseamar's pilots due to the gliders sensitivity to water currents. Defined in 4D (3D + time), they play a key role in the mission preparation and supervision phase. However, it is important to notice that the spatial and temporal sampling of these data are different from the effective sampling time of the glider.

Finally, before starting a mission, a consumption model of the SeaExplorer glider is used in order to compute the total number of segments achievable. A Battery Management System (BMS) is also embedded on the battery pack of the glider. It allows pilots to retrieve information about the battery level during the mission through the mission management system. The mission plan can be adapted according to the remaining energy. Regardless of the number of vehicles used over the operation area, the drop point and the points to be targeted are defined before the start of the mission. Fig. 7 depicts the three functional levels of behavior of the glider relative to its current mission (planning), relative to its environment (navigation) and relative to its own internal states and its actuators to make them change (control). The last level dedicated to the glider characteristics is detailed in Section 2.1.3. The navigation level deals with the flight profile and the heading to be targeted to hit the next waypoint. Presently the navigation task is assured by the pilot who has to manually define the coordinates of the next waypoint but also to manually adjust the flight profile to take into account the maximal depth given by the bathymetric product. Above this level, the planning level is also assured by the pilot who has to manually edit the list of the next waypoints to hit. These target waypoints may also be provided from high level path planning algorithms.

3. Simulator specifications

The development of this full mission simulator is based on the in-depth understanding of what was previously described. It is designed in the same way as the current control of the SeaExplorers (piloting through the GLIMPSE mission management system). It implies that it takes into account the same input data in order to be able to test the outputs of high level algorithms for the generation of single-vehicle and multi-vehicles trajectories. It also means that the important features of GLIMPSE are an integral part of the simulator: calculation of the heading of the glider for the target waypoint (Section 2.2), possible modification of the navigation profile (Section 2.1.1),

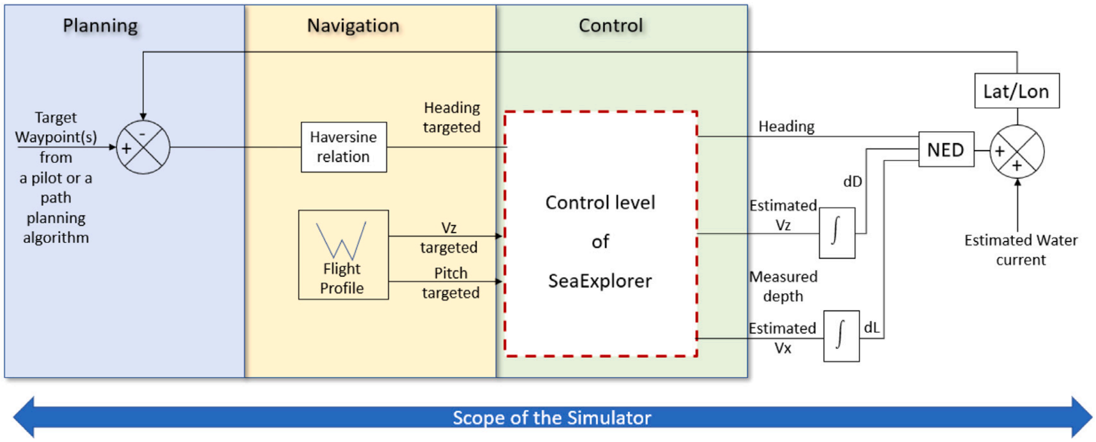


Fig. 7. SeaExplorer functional levels (Planning, Navigation, Control).

activation/deactivation of sensors (payload profile) (Section 2.2), activation/deactivation of the water current compensation (Section 2.1.4). It is the reason why the development of the simulator is based on the GLIMPSE "mission plan" mode. This mode is crucial for the glider mission simulation in both single and multi-vehicles configurations. In this mode, the vehicle has to validate numerous waypoints according to a given navigation profile. It also includes other features that are not available yet via GLIMPSE but which are of great importance for the running of a SeaExplorer full mission: hydrodynamic (Section 2.1.3) and consumption models (Section 2.2) of the glider, environmental modeling (4D water current and 3D bathymetry levels), near "real-time" management of fleets.

Due to the great distances that SeaExplorer gliders can have to travel, it is important that the simulator allows generating the trajectories on the geographical plane and not only on the Cartesian plane. The shape of the Earth and the geographical position of the simulated drone may affect the accuracy of the trajectory (this aspect will be further detailed in the next section). The aim of this simulator is to match as closely as possible to the current way of piloting gliders from GLIMPSE. From this platform, it will be possible to interact with higher level algorithms for the management of SeaExplorer underwater glider fleets. Our simulator is coded in Python object language for compatibility reasons with the embedded code of the platform.

It must therefore have a fairly low computational cost compared to the duration of a glider "yo" (which lasts around 3 h). An adaptive sampling time is calculated in the simulator to reduce the computational cost. It consists of computing the time step size required for the simulated glider to reach the immersions available in the used water current dataset. The simulator also provides a Human–Machine Interface (HMI). It allows an easy access to the various graphs displayed from the GLIMPSE mission management system when piloting a SeaExplorer:

- The trajectory of the vehicle in the Cartesian and geographical coordinate system,
- The battery level which decreases as a function of time,
- The consumption of the navigation and payload parts which vary as a function of time,
- The targeted and the followed heading,
- The navigation phases,
- A display of different metrics such as the total time of the simulated mission, the distance traveled by the simulated vehicle, the energy still available at the end of the mission.

In addition to meeting these requirements, the simulator has to include the three functions described below:

1. For a given path to follow and payload configuration, the simulator estimates the time to travel, the continuous autonomy and the accuracy to stick on the desired path. So for the preparation phase, the simulator is used in direct mode to aid the pilot to make a decision relative to the best route to follow and to the best payload configuration (**planning phase**).
2. During the mission, the simulator assists the pilots in their **navigation tasks** for the definition of the best flight profile and path to adopt by integrating the surrounding bathymetry and water currents given by the dedicated maps.
3. Regarding the path planning, the simulator can integrate the bathymetry and the behavior of the controlled glider by taking into account the surrounding environment (water current given by satellite maps) to accurately assess the cost to travel from a point A to a point B. The simulator is a powerful tool to fill adjacency matrices due to its calculations of the duration, the travel distance and the energy consumption. It enables to measure the cost between any points and their neighbors and thus the edges connecting the vertices of a weighted graph (Bondy and Murty, 1976). These graphs are used to model the tackled problem and the adjacency matrices are the representation used to minimize the cost function in high level path planning algorithms. Moreover the simulator can evaluate the efficiency of these algorithms. It means that in one hand it provides high level algorithms with important data relative to the travel cost but also it helps to compare the different strategies (paths) to meet the requirements of a mission. It is a substitute for the pilot in the planning phase but also in the navigation phase. The aim is to tend to a full automatic mission controller for one glider as well as a fleet.

The specifications and the span of the simulator have been detailed, its architecture is explained in the next section.

4. Simulator framework

This simulator is coded in Python Object and encompasses several files corresponding to the different classes that were developed. The global architecture is structured in three levels called "Mission level", "Dive level" and "Glider level". The "Mission level" is centered around a "Mission simulator" which is supplied by inputs from the users or high level algorithms. Most parameters of interest are stored in the "Logs" class, which possibly interacts with the HMI in order to provide a visualization of the important metrics and graphs (as detailed in Section 3) to help the end-users in their choice and validation of a given mission plan. This "Mission simulator" main function manages the chaining and validation of a sequence of waypoints. The latter are

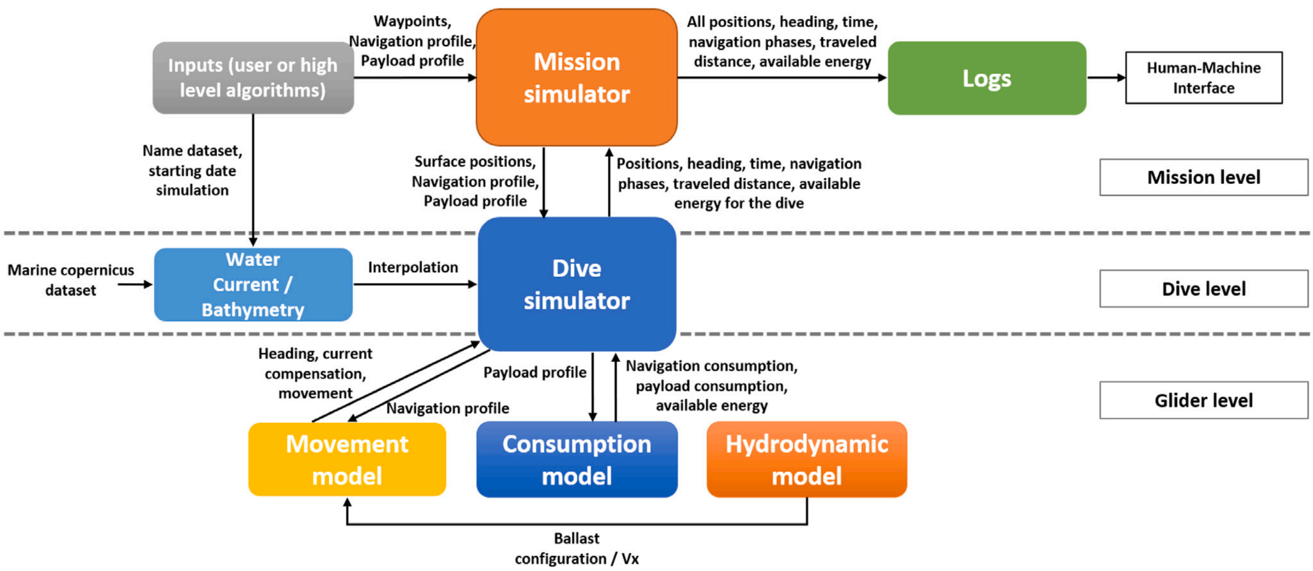


Fig. 8. Overview of the simulator framework.

considered as hit when the distance between the geographical position (latitude, longitude) of the glider and the target waypoint is shorter than the validation distance. Concerning the chaining, the function calls, as many times as necessary, the “Dive simulator” routine. This function is used to calculate the trajectories of a SeaExplorer during a dive taking into account an initial surface position and a “Water current” model of the area given as input of the “Mission level”. These two parts compose the second level named “Dive level”. This latter interacts with the “Glider level” which is the lowest level. It takes into consideration all the features related to one given vehicle such as the “Movement model”, the “Consumption model” and the “Hydrodynamic model” that will be presented in the next subsections. It ensures the modular aspect of the simulator which could be applied to different types of vehicles by only changing the models on this third level.

The synoptic view of the functional principle of the simulator is provided in Fig. 8.

Now the main functional blocks are further detailed.

4.1. Mission definition (first level)

A glider mission simulation requires the definition of a “Mission” object as input of the “Mission level” and particularly as input of the “Mission simulator”.

The definition of this object is similar to the “mission plan” mode as explained in Section 2.2. It takes as inputs a list of target waypoints (wp) in the format: $[Lat_{wp}, Lon_{wp}, V_d]$, where Lat_{wp} (resp. Lon_{wp}) stands for the latitude (resp. the longitude) of the target waypoint and V_d the validation distance. It allows defining the navigation profiles with different flight parameters that the vehicle has to adopt during a simulated segment: $[V_z, \alpha, z_t, z_b, z_s, sr, t_s]$ where t_s stands for the time to wait during a surfacing. These parameters have been defined in Section 2.1.1.

The payload profiles are also configurable through the definition of the states of activation/deactivation of the altimeter, the radio and the payload. Finally, a “segment” attribute constitutes a crucial parameter for the definition of a glider mission since it specifies a sequence of waypoints by setting a list of initial waypoints, waypoints to target with the flight profile and payload numbers considered for each segment to be carried out by the simulated glider. Such a format was adopted to correspond to the “mission plan” module implemented on the “GLIMPSE” mission management system. This facilitates the use of the simulator by SeaExplorer drone pilots and the simulation of the lists of waypoints generated by high level path planning algorithms.

At the start of the mission simulation, a “AskUser” class as part of the “inputs” of the first level, requests information from the user in the prompt. The user can then choose to use a water current file and a bathymetry file in the “NetCDF” format (Network Common Data Form, format generally used for multidimensional scientific data) available either locally on the user’s computer from which the simulation is launched or from a “Files Transfer Protocol” (communication protocol intended for file sharing on a TCP/IP network). The latter fetches products in “NetCDF” format directly from the Marine Copernicus database. Once the product has been selected, the initial date has to be defined to synchronize the start time of the simulation with the water current dataset. The water current products available from Marine Copernicus provide access to the East and North components of the measured or modeled water current as well as its position $(Lat_{current}, Lon_{current})$ and its immersion. The bathymetry products also available from Marine Copernicus provide access to the bathymetry levels depending on the geographical position.The management of the water current and the bathymetry levels on the area of the simulated glider is performed by a function of the “Mission” class detailed in the next section.

Finally, the developed simulator uses multi-threading to generate trajectory simulations for several gliders with different configurations and missions in the multi-vehicles mode.

4.2. Environmental modeling (second level)

Water current information from the models can be modeled as “water cubes”. The latter are available at several dates depending on the Marine Copernicus product that is considered.

SeaExplorer trajectory generation through the “Glider level” can result in positions that are not perfectly on a $(Lat_{glider}, Lon_{glider})$ position and at a depth existing in the water current dataset. It is the reason why a method of the class “Mission” enables to generate several types of interpolation on a 4D grid: time, latitude, longitude, immersion. Nevertheless, the nearest neighbor interpolation is the cheaper in terms of computational time.

It is important to note that in most of the water current datasets available, the spatial sampling is not constant, in particular for the “immersion” vector. Initially, the simulator was planned to operate with a constant time step of 20 s, since it corresponds to the sampling time of the glider. However, in order to decrease the computing time, an adaptive time step size was set up (parameter adjustable in the definition of the “Mission” object).

This adaptive time step size consists of calculating the step size necessary to reach the different immersions available in the water current dataset while taking into account the maximum, the minimum and the surface immersions ordered. To do so, distances between the immersion of the glider and the immersions of the dataset are computed and the different values of the adaptive time step size are calculated by dividing the distances by the vertical speed of the simulated glider. This adaptive time step size is limiting the number of positions to be calculated during the dive of the simulated glider. Note that when it comes to the simulation of the most energy consuming navigation phases (ballast inflection phases), the time step size is set to 20 s to ensure accuracy in the consumption model. This so-called adaptive time step size considerably reduces the simulator calculation time: for a simulation of 400 waypoints (square grid with sides of 200 km to cover), the constant time step size of 20 s leads to compute the trajectories in 32 s while it only takes 4 s with an adaptive time step size (tested on an Intel(R) Core(TM) i7-9750H CPU, 16 GB of RAM). This "NetCDF" products based management of the water current in the area of operation means a different way to model the environment than other existing simulators evoked in the Introduction 1.

As part of the water current model selected, it is also possible to have access to the bathymetry levels of the Marine Copernicus product. At each time step, interpolations are generated to get an estimation of the bathymetry level at the geographical position of the simulated glider. During the simulation, the maximal immersion commanded is modified according to the interpolated values in order to take into account the seabed.

The interpolated data of water current and bathymetry levels supply the "Dive simulator" which sends the desired navigation profile to the lowest level named "Glider level". It calls methods of the three models belonging to this level in order to generate the glider trajectories during a dive and to send the major parameters that are calculated, back to the "Mission level".

4.3. Glider level (third level)

We now describe the lowest level which is specific to the vehicle that we consider. In our case, we focus on the SeaExplorer.

4.3.1. Glider trajectories modeling

The flight profile parameters are received as argument by the "Movement model" class, which also takes into account a desired heading calculated beforehand by a method of the "Mission" class (using Haversine formulas (2002)). Simultaneously, the ballast configuration of the glider for the dive is provided with a knowledge-based model of the "Hydrodynamic model" class as a function of the vertical speed and the target pitch. The hydrodynamic model of the glider is directly incorporated into this class as it is actually done on the SeaExplorer glider (as detailed in 2.1.3).

The "Movement model" class therefore requires a call from the "Hydrodynamic model" of the vehicle to define the ballast configuration that the simulated vehicle must adopt and to have an estimation of its horizontal speed. This motion class is used to describe the behavior of the SeaExplorer drone. Therefore, a certain number of methods are available to generate its trajectories. Note that the Forward Euler discretization method is used in the simulator. Some of the methods also allow a change in the coordinate system points between the vehicle coordinate system (see Fig. 3), the NED coordinate system (North-East-Down Wiley and Sons, 2011) and the geographical coordinate system. It is possible to represent the NED coordinate system as displayed on Fig. 9.

The change of coordinate system between the glider and the NED, is given by Eq. (1):

$$\mathbf{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} \cos(H^{(n)}) & \sin(H^{(n)}) & 0 \\ -\sin(H^{(n)}) & \cos(H^{(n)}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} N \\ E \\ D \end{pmatrix} \tag{1}$$

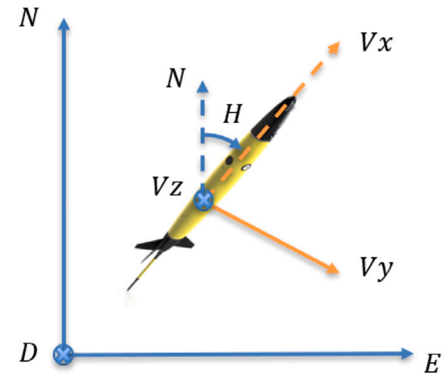


Fig. 9. Static resolution and NED coordinate system.

where $H^{(n)}$ is the current heading clockwise from North of the simulated glider at time iteration n.

To reach a desired heading, the heading is calculated thanks to Eq. (2):

$$H^{(n)} = H^{(n-1)} + \frac{V_x}{T_r}.\mathrm{d}t \tag{2}$$

with dt the time step size and T_r standing for the "Turning radius" parameter i.e. the gyration radius taken by the vehicle when it dives to reach a calculated heading. This parameter is fixed at ±50 m to turn left or right and is integrated in the "Movement model" class. It is important to notice that the initial surface heading is randomly generated to simulate an orientation of the vehicle as a function of the wind present in the operation area. The motion of the SeaExplorer is described by the following system of equations in the NED coordinate system (3):

$$\begin{cases} \mathrm{d}L = V_x.\mathrm{d}t \\ \mathrm{d}N = \cos(H^{(n)}).\mathrm{d}L \\ \mathrm{d}E = \sin(H^{(n)}).\mathrm{d}L \\ \mathrm{d}D = V_z.\mathrm{d}t \end{cases} \tag{3}$$

with $\mathrm{d}L$, the length of the 3D displacement during the considered time step.

From the management of the heading of the glider and the calculation of its displacement on the Cartesian plane, a method was developed in order to switch from the NED coordinate system to the geographical coordinate system thanks to the Haversine formulas (4):

$$\begin{cases} Lat^{(n)}_{glider} = \arcsin(\sin(Lat^{(n-1)}_{glider}).\cos(\frac{\mathrm{d}L}{E_r}) \\ \qquad + \cos(Lat^{(n-1)}_{glider}).\sin(\frac{\mathrm{d}L}{E_r}).\cos(H^{(n)})) \\ Lon^{(n)}_{glider} = Lon^{(n-1)}_{glider} + \arctan 2(\sin(H^{(n)}).\sin(\frac{\mathrm{d}L}{E_r}).\cos(Lat^{(n-1)}_{glider}).\cos(\frac{\mathrm{d}L}{E_r}) \\ \qquad - \sin(Lat^{(n-1)}_{glider}).\sin(Lat^{(n)}_{glider})) \end{cases} \tag{4}$$

where E_r stands for the Earth radius and $Lat^{(n-1)}_{glider}$ (resp. $Lon^{(n-1)}_{glider}$) is the latitude (resp. longitude) value from the previous iteration. The Haversine formulas take a spherical planet Earth as a model, which may imply precision errors up to 0.3% over fairly large distances. Within the simulator, all the calculations are carried out over a low time step size and therefore over fairly short distances implying that the simulator is less sensitive to such errors even in the adaptive time step size mode.

The trajectories obtained with the equations detailed above do not take into account the influence of the sea currents on the glider trajectories. This is why a notion of current has been integrated into the simulator in the "Mission" class (as detailed in Section 4.2). The "Movement model" class uses the water current information available from the 4D interpolation, namely $E_{current}$ for the East component and

$N_{current}$ for the North component of the water current speed (V_c) as well as the direction of the current (D_c) for its two directions clockwise from North, to calculate a drift due to the water current in the NED coordinate system (5):

$$\begin{cases} \mathrm{d}N_{current} = V_c.\mathrm{d}t.\cos(D_c) \\ \mathrm{d}E_{current} = V_c.\mathrm{d}t.\sin(D_c) \end{cases} \tag{5}$$

Then, it is possible to have the target trajectory of the vehicle (named "dead-reckoning trajectory") and the trajectory under the influence of the water current (called "current-corrected dead-reckoning trajectory") both on the Cartesian plane and on the geographical plane.

Finally, a method of the class calculates the magnitude and direction of the average water current perceived by the glider during the previous dive. It is done by calculating the difference between the surface position of the "current-corrected dead-reckoning trajectory" and the surface position of the "dead-reckoning trajectory" as well as the bearing difference between these two positions. Another method allows compensating this average water current perceived by the vehicle in the calculation of the target heading. It is presently implemented on the SeaExplorer drones (see Fig. 10).

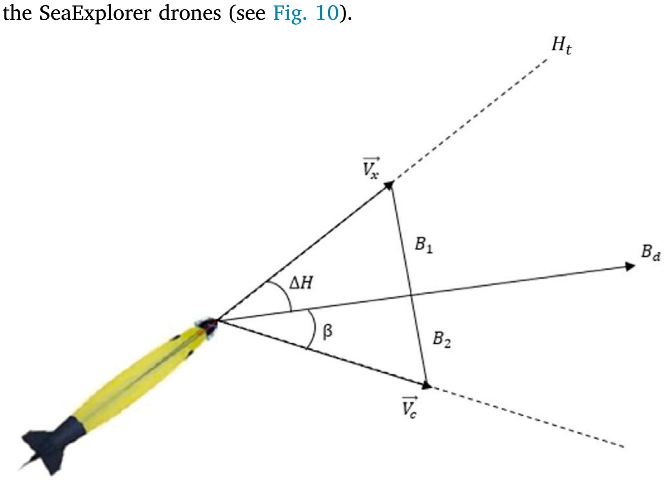


Fig. 10. Compensation of the water current.

If the current is not taken into account, instead of following the desired bearing (B_d), as shown on Fig. 10, the glider follows the target heading (H_t) which is assimilated to the desired course.

The method consists of calculating the heading correction (ΔH) to follow a desired bearing, according to Eq. (6) in the NED plane:

$$B_1 - B_2 = 0 \Leftrightarrow V_x.\sin(\Delta H) - V_c.\sin(\beta) = 0 \Leftrightarrow \Delta H = \arcsin\left(\frac{V_c.\sin(\beta)}{V_x}\right) \tag{6}$$

The updated desired heading corresponds to the previous one (calculated using the "Mission" class) to which is added the ΔH calculated. The simulated vehicle is then able to compensate the water current perceived on the previous dive as a real SeaExplorer would (detailed in Section 2.1.4). With these equations, it is possible to generate a displacement of a SeaExplorer in the NED coordinate system and to use it for the management of the navigation phases of the glider. As described in Section 2.1.2, the navigation phases depend on the vertical displacement of the vehicle with surface, maximum and minimum immersion instructions. The glider must respect these instructions when it performs its navigation cycle in the water column. However, there are certain phases (110, 115, 118) during which the vehicle does not have any immersion instruction to follow (as described in Section 2.1.2). For the simulation, the duration of each phase is calculated from the ballast volume and the oil pump efficiency controlled in flow.

The following system of Eqs. (7) is considered:

$$\begin{cases} T_{110} = \frac{B_s - B_d}{I_d} \\ T_{118} = \frac{B_u - B_d}{I_u} \\ T_{115} = \frac{B_s - B_u}{I_s} \end{cases} \tag{7}$$

with B_s standing for the ballast configuration at the surface (fixed value at 500cc), B_u and B_d being the up and down ballast configurations given by the hydrodynamic model as a function of the target vertical speed and the target pitch. I_d (resp. I_u) is the value of the inflecting down (resp. up) pump flow. I_s is the value of the inflecting up pump flow to the surface.

4.3.2. Energetic consumption modeling (third level)

The interest of such a mission simulator lies not only in playing upon the flight profiles but also in comparing the main metrics of different scenarios (i.e. duration of the mission, total travel distance, remaining autonomy).

The challenge is to optimize the mission according to these three output parameters while keeping in mind that the battery level is the most critical one (a deep discharge below 15% must be avoided). To do so, the "Consumption Model" class finely includes the payload consumption as defined in the "Mission" class. A "payload_config" member variable specifies the type of the embedded sensors on the simulated vehicle (18 sensors are predefined and selectable) with their activation modes. The type of battery pack equipping the glider is also specified (4 battery packs are available).

The various energy consumption parameters associated are available in a parameter file which includes:

- The power values of the sensors,
- The power measured on several vehicles for the different navigation phases,
- The available energy values for the different battery packs.

Thus, a method has been developed to determine the energy consumption during the simulated mission of the navigation part and the payload part of the drone.

The energy consumption varies according to the navigation phase in which the glider is. For example, it costs more to activate the ballasts when the vehicle is at depth rather than on the surface due to the pressure exerted on the vehicle. A representation model is used in the simulator to get the values of the power relative to the ballast and the immersion. This model is also used to get the power value relative to the pitch during the transitions phases, the power value associated to the roll being a constant value during the turning of the simulated glider. It is important to note that the simulated heading and pitch are considered as invariant when the target heading and pitch are reached. The consumption of the servoing is considered as negligible in this simulator. The developed method therefore strongly depends on the management of the navigation phases of the glider in the "Movement model" class. It takes into account the navigation phase in which the drone was in the previous step and calculates a consumption allocated to the navigation part of the vehicle by multiplying the time step size by the power associated with the navigation phase.

For the part allotted to the optional sensors of the glider, namely the altimeter and the radio, it is necessary to calculate respectively an energy consumption depending on the phase of the dive (because the altimeter is only active during the dive) and a another depending on the communication phase on the surface (the radio is only activated when the vehicle starts to communicate).

For the sensors equipping the simulated SeaExplorer, it is possible to define their activation states (e.g. if they are needed during the descent or the transition phases or the ascent). Thus, the energy consumption of this part depends on the activation states of the sensors (which can vary through the navigation phases). For example, if a sensor is only

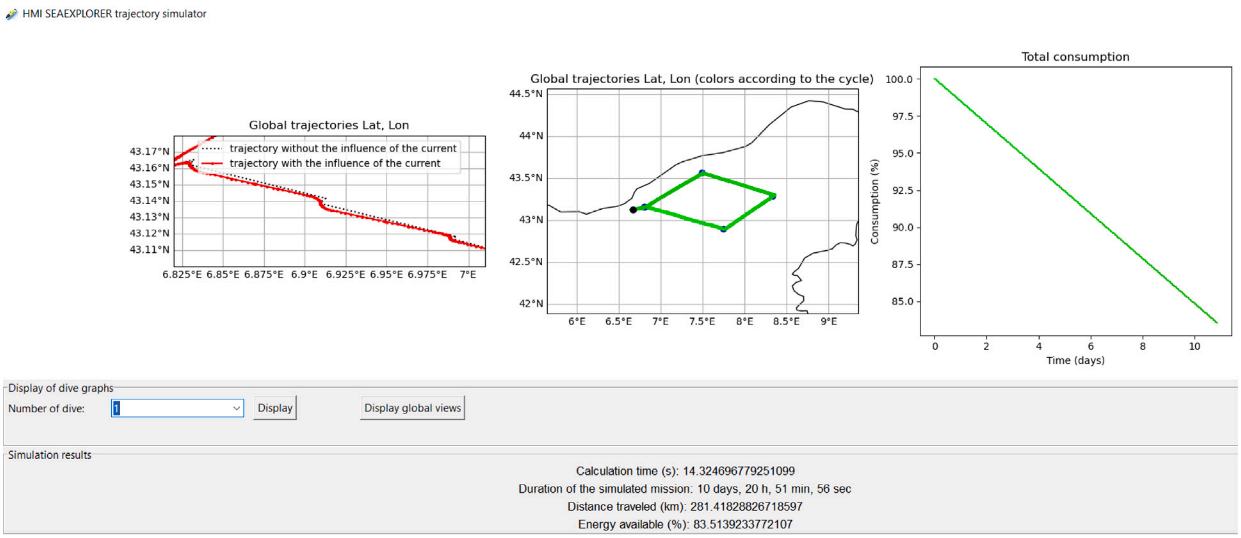


Fig. 11. Global view of the HMI. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

activated during the descent, it is necessary to multiply the power of the sensor by the time step size if the vehicle is in phases "Going down" (100) and "Inflecting down" (110). If the sensor is on during phases of transitions, it will be necessary to multiply the power of the sensor by the time step size of the "Transmitting" phase (116). If the sensor is activated during the ascent, it is necessary to multiply its power by the time step size calculated if the drone is in phase "Going up" (117), "Inflecting up" (118), "Surfacing" (115).

Thanks to this method, it is possible to calculate the energy consumption of the navigation part, of the optional part and of the sensors for each time step, taking into account the navigation phase of the drone. The total energy consumption is given by the sum of all these contributions to energy consumption. The available energy is the difference between the energy available from the battery pack used and the total energy consumption. So the current available energy is monitored all along the mission.

4.4. Outputs through a Human–machine Interface (first level)

During the mission simulation, the "Logs" class stores all the variables computed by the simulation: the glider positions in the Cartesian plane, in the geographical plane taking into account (or not) the influence of the current, the desired headings, the headings followed by the simulated glider, the navigation phases, the simulation time vector, the energy consumption, the available energy, and so on. This class allows to have a detailed follow-up of the entire mission simulation and to have access to the important outputs. The HMI developed for our simulator is based on this class and on the import of the `Tkinter` python graphic library. It provides access to the global trajectories on a geographical plane with and without the influence of the water current on the glider over its entire mission (zoomed in Fig. 11). The global path is drawn in a color that depends on the level of the available energy on the segment (green level for 100% of battery, black level for 15% of battery or less). The available energy versus time is displayed on a third chart (see Fig. 11).

Moreover, the end-user can have a look at the trajectory carried out by the simulated glider in a 2D or 3D Cartesian plane for any dives (see Fig. 12) and monitor the navigation phases. On the 3D trajectory, it is possible to see red dots corresponding to the bathymetry levels interpolated as explained in Section 4.2. Finally, the HMI also provides a direct access to the important metrics: computational time taken to simulate a mission, time taken by the vehicle to carry out the simulated mission, travel distance and percentage of available energy at the end of the mission.

The objective of the next section is to validate the simulator by confronting the results of a real experimental mission with those obtained thanks to the same simulated mission.

5. Experimental validation and discussion

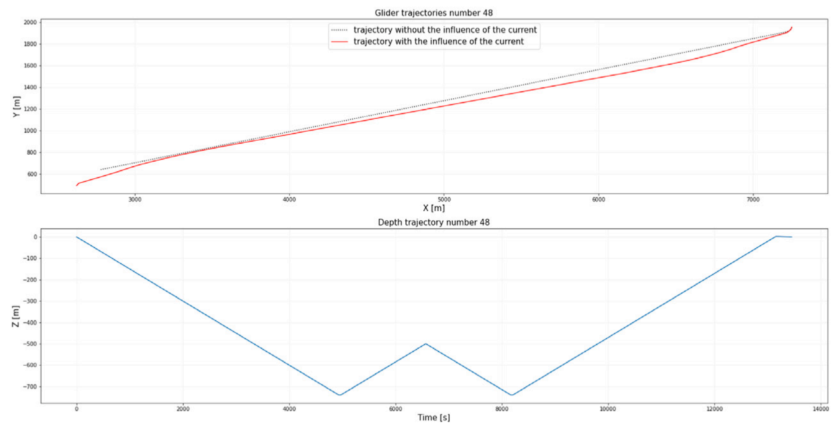
On January 19, 2022, Alseamar has led a mission where a SeaExplorer was deployed on the "MOOSE T00" transect ("DYFAMED" series) between Nice (French Riviera) and Calvi (Corsica island). The aim of this mission was to "better understand the mesoscale variability of the hydrological and biogeochemical processes of the Ligurian Sea" (Laurent, 2008). This mission is used as ground truth to validate and evaluate the accuracy of our simulator. The SEA041 mission has been simulated with the same parameters as those used in the real mission. The mission was composed of 2 returns trips between Nice and Calvi and one single trip to Calvi (see Fig. 13 for a snapshot of the monitoring of this mission through the mission management system). The drone was operated with 20 degrees of pitch at 0.20 m/s while doing 0–1000 m dive in the water column with only one "yo" and a "z_s" of 10 m. The embedded sensors were a "FLBBCD" fluorescence sensor, a "GPCTD" (pumped conductivity measurement, temperature and depth) sensor, a "RBR" (CTD) sensor and a "RinkoDO" oxygen sensor.

These flight parameters and payload configurations were therefore considered as inputs for the simulator using a 4D water current model (Water current model, 2015) of the area in order to generate the most realistic simulation of the SEA041 mission.

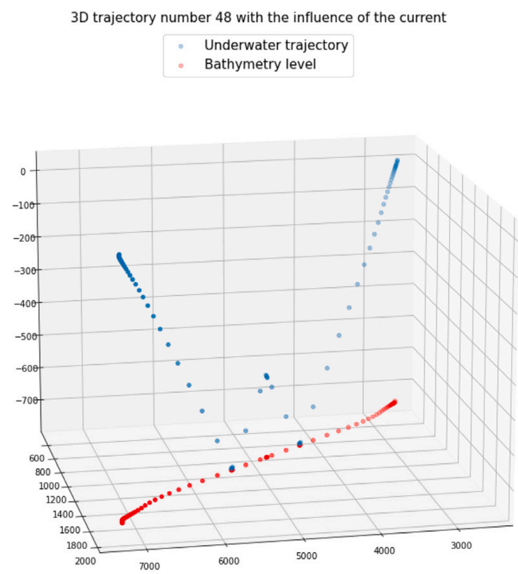
The simulated glider performed 113 dives versus 116 dives (see Fig. 14) during the real mission with 61.92% battery remaining (versus 59.85%).

This means that there is approximately 2.6% error on the number of dives carried out between the simulation and the real marine experimentation and approximately 1% error on the remaining energy at the end of the mission for an equivalent number of dives (see Fig. 15 for a remaining energy comparison between the SEA041 mission and the simulation).

These disparities may be due to certain events that occurred during the piloting of the drone SEA041 such as a late activation of the water current compensation during dive number 15 (while it was activated from the start in the simulation) and the activation of the altimeter when the vehicle approached the coasts (whereas it was deactivated the whole time in the simulation).The results obtained from a trajectory point of view remain consistent with the real mission (as depicted in Fig. 16 for a given transect) despite the fact that the simulated



(a) Cartesian plan trajectory



(b) 3D trajectory

Fig. 12. 2D and 3D trajectories of the simulated glider. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

glider is faster (because of the activation delay of the water current compensation during the real mission).

This mission validates the relevance and efficiency of the simulator as a powerful tool that can be used in mission preparation and also in the supervision phase. It takes only about 28 s to generate such a mission. The simulator can highlight the influence of several parameters of the flight profile such as the maximal depth or the number of "yos". For instance, decreasing the number of dives down to 79 instead of 113 as previously, the simulation shows in Fig. 17 that the same mission can be achieved sooner (12 days, 10 h, 46 min instead of 12 days, 15 h, 6 min) but the consumption is higher than before (56.16% battery left versus 61.92%).

The simulator can also be used for the composition study of a payload during the preparation phase of a mission. For example, it is possible to play as many scenarios as needed with different list of sensors in order to ensure the validity of a mission plan. Fig. 18 shows the outputs of a SEA041 mission simulation with four more sensors equipping the simulated glider: an acoustic sensor, a methane sensor, a turbulence sensor and nitrate sensor.

With this payload configuration, the energy available at the end of the simulated mission is around 14.49%. As stated before, a battery level below 15% is critical for the recuperation of the glider thus this payload configuration is not a reasonable option for the mission. A summary of these tests can be find in the Table 2:

Table 2
SEA041 mission and simulation outputs comparison.

	Real mission	Simulated mission	Simulation with $sr = 2$	Simulation with a different payload
Dives number	116	113	79	113
Energy available	59.85%	61.92%	56.16%	14.49%

6. Numerical path planning evaluation

The interest of the simulator is also to be able to test the trajectories obtained by higher level algorithms in different types of "routing problems". Among these problems, we focused on the "coverage path planning" which consists of finding an optimal path passing through all the points of a predefined zone of interest. This type of coverage can be applied in the world of underwater gliders. For example, on oil and gas missions (Meurer et al., 2021), the goal is to perform a measurement acquisition in depth in each of the boxes of a predefined zone in order to detect a possible hydrocarbon seep which could indicate the presence of a deposit.

As it was showed in the previous section, the results of a simulated mission are very close to those obtained during the real mission, that confirms the relevance of our simulator. Moreover the simulator plays

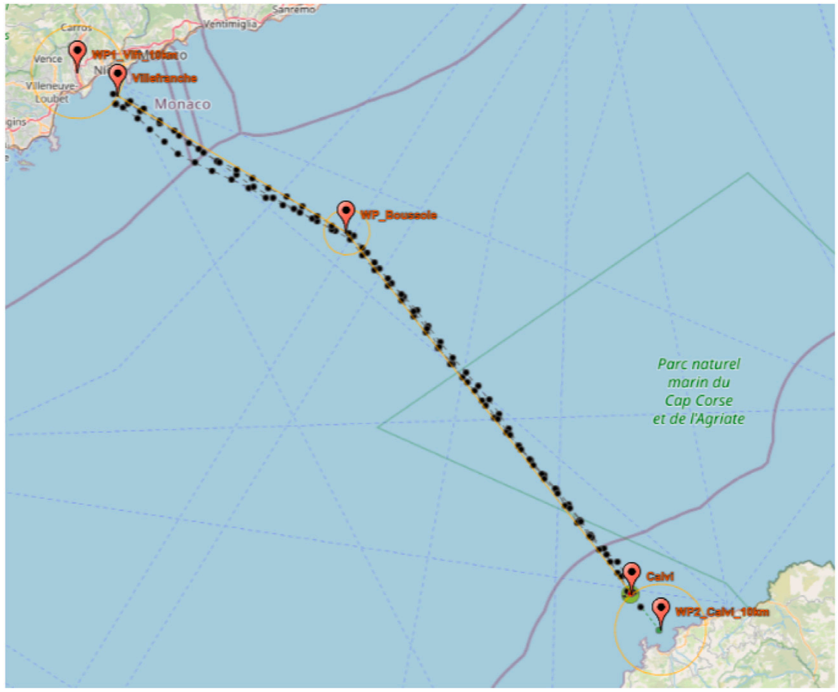


Fig. 13. SEA041 mission viewed from the mission management system.

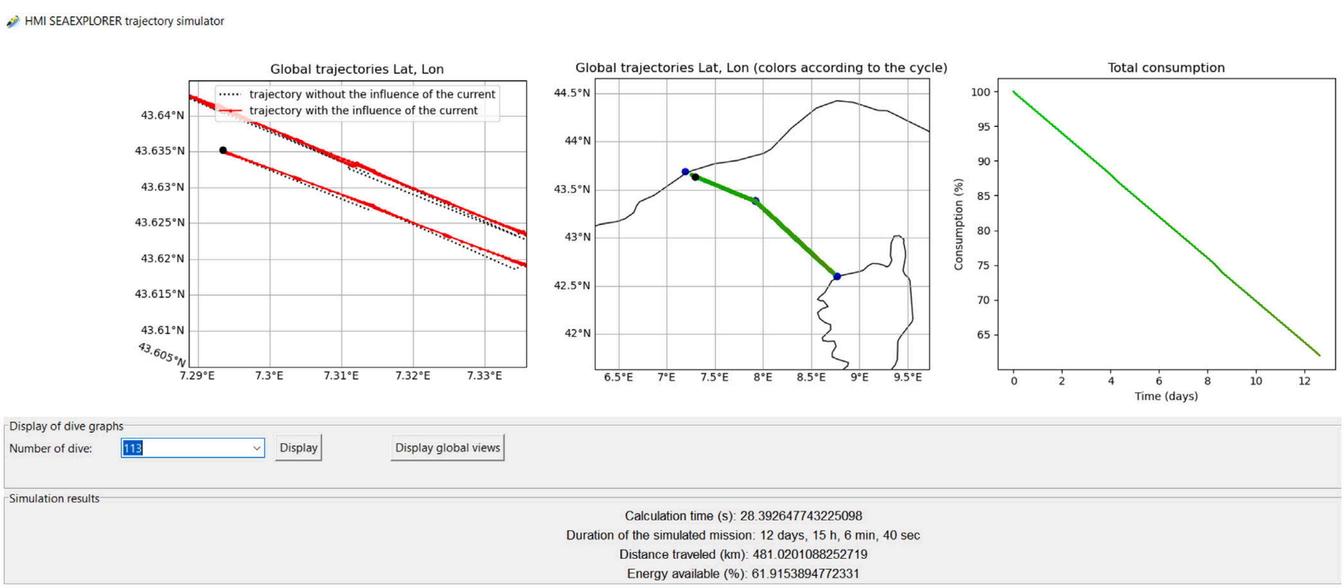


Fig. 14. Output of the SEA041 mission simulation, 113 dives.

a huge interest role in path planning because it provides the values of the main metrics of any paths on the considered map. Actually, based on the behavioral model of the whole glider and the model of the environment (3D bathymetry and 4D water current), the simulator gives the travel time, the energy consumption, the simulated path for several flight profiles between any waypoints and its neighbors. It opens up the opportunity to fill adjacency matrices of weighted graphs which are the representation used to minimize the path cost relative to the criterion (distance, time, energy) used by high level path planning algorithms.

Beyond getting the optimal path (planning level), the use of the simulator contributes also to improve the navigation task by evaluating the efficiency of different flight profiles. Indeed to avoid too strong water current in low depth, it is sometimes more cost-efficient to keep moving forward in higher depth. Such a way, once the mission preparation phase is completed, the optimal path is transferred to the mission management system GLIMPSE with a flight profile adapted all along the path. It is in this line of thought that some external path planning algorithms have been developed. One of the preliminary work that we carried out was to seek for a Hamiltonian path (Rahman and Kaykobad, 2005). The goal is to converge towards a path that goes through all the nodes of a predefined graph exactly once. For the simulation, an area to be covered was selected between Nice and Calvi. This area is made up of 30 nodes to be visited by a glider, these nodes constituting waypoints with a validation distance of 2500 m.

The developed algorithm allows to take into consideration an adjacency matrix built automatically by taking into account the connections

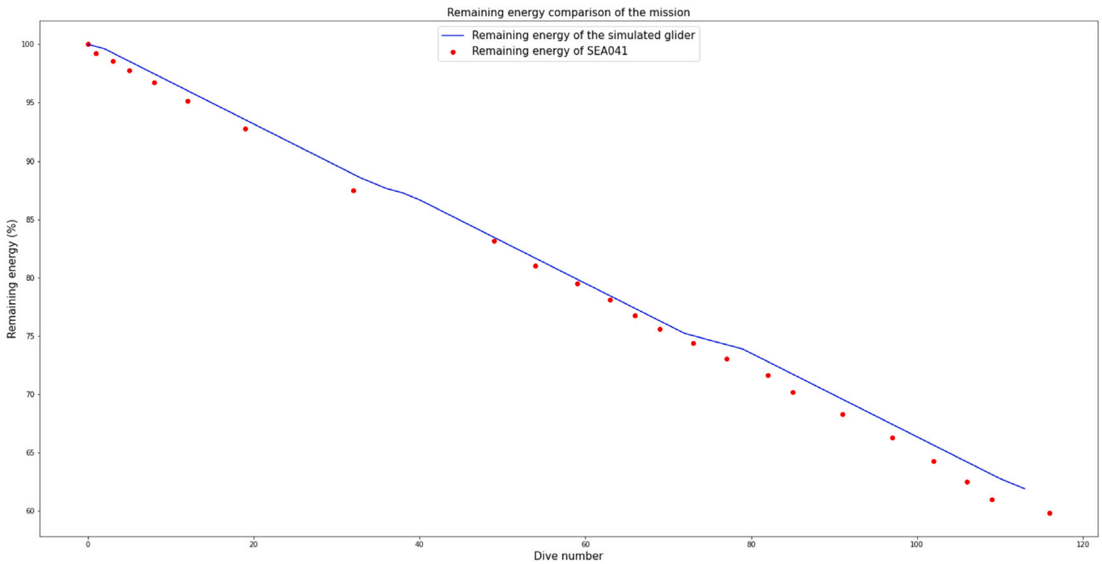


Fig. 15. Remaining energy comparison between SEA041 mission and simulation.

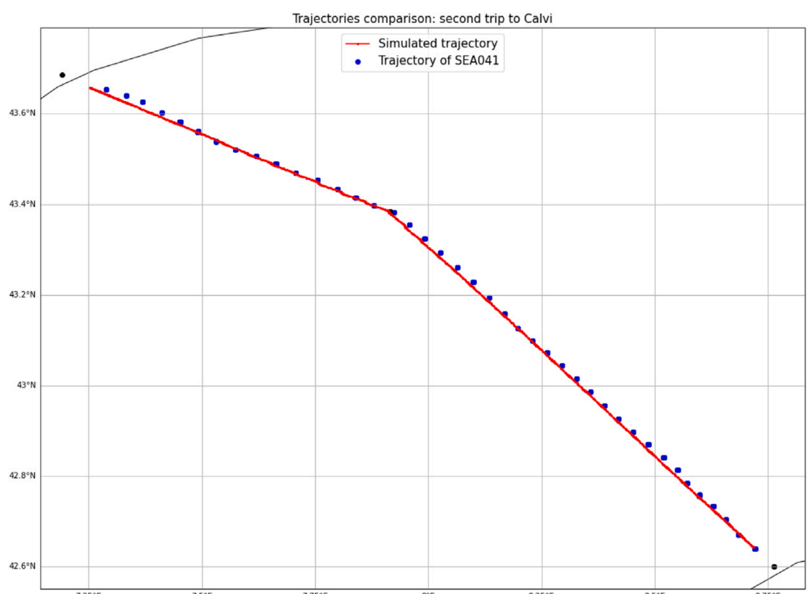


Fig. 16. Trajectories comparison between simulated and real mission on a given transect.

in neighborhood "4" between the nodes of the zone of interest (connections in row and in column only) and an initial point of deployment that was fixed. An order of succession of the nodes ensuring a Hamiltonian path is then constructed. This order of succession is used to define a mission object within the simulator. Thus, it is possible to use the simulator in addition to this algorithm to validate the generated trajectories.

By playing upon the initial deployment point, we have built different trajectories with the Hamiltonian path planner (Fig. 19(a) and (b)).

In this example, there are substantial gaps on travel distance, 20.29 km, and on energy consumption, 1.43%, as indicated in the Table 3. During this simulation, the same flight profile as for the mission of the SEA041 glider was used. This latter takes place in a part of the Liguro-Provençal water current between Nice and Calvi and uses the same 3D water current dataset as for the simulation of the SEA041 mission.

On Fig. 19, the surface water current flow fields (even though the simulator uses a 4D model) is represented by vectors of different length depending on their intensity, and their orientation is given by their North and East components. The Table 3 shows that the water current is quite significant and it may have a major impact on the trajectories of the simulated vehicles and in particular on the metrics of mission duration, the travel distance and the energy available at the end of the mission. From such outcomes, a "Lawn mower's path" starting from the North of the area is a better option than starting at the South-West due to the strong descending water current from North-East to South-West. This difference in available energy may seem small at first glance, however it is important to keep in mind that SeaExplorer drones are generally used during long missions of several months and this difference can become really significant over time. So we can see that the point of deployment may play a determinant role in the success of the mission, this emphasizes the interest of our simulator in choosing the best trajectories and the most appropriate high level path planning algorithms.

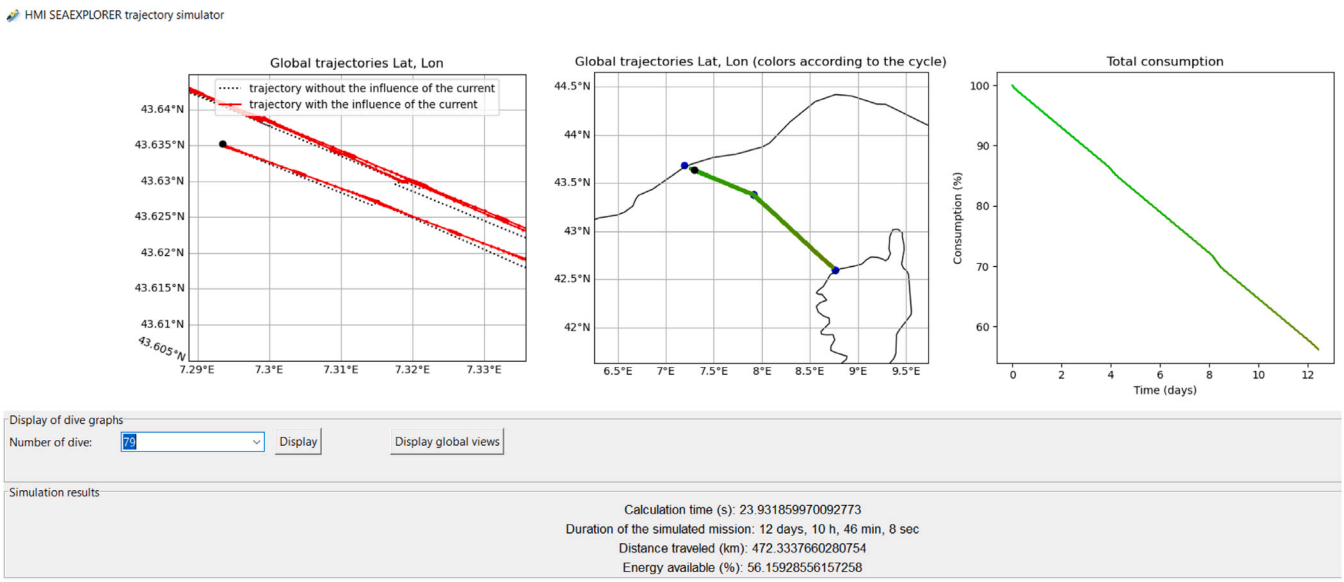


Fig. 17. Output of the SEA041 mission simulation with sr = 2, 79 dives.

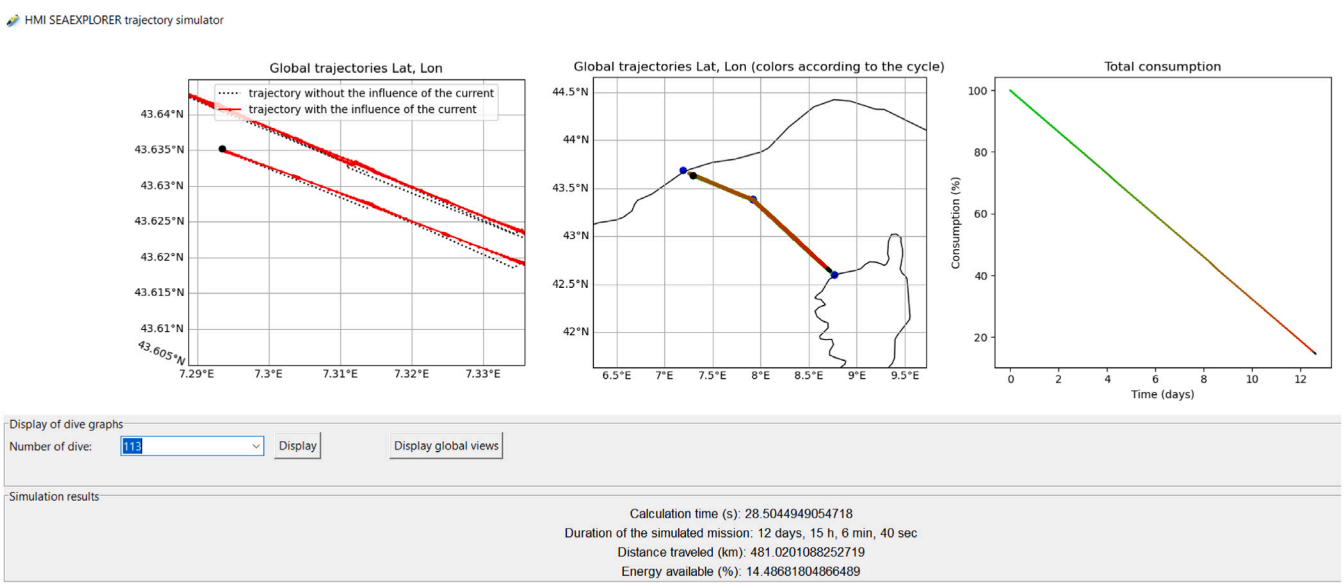


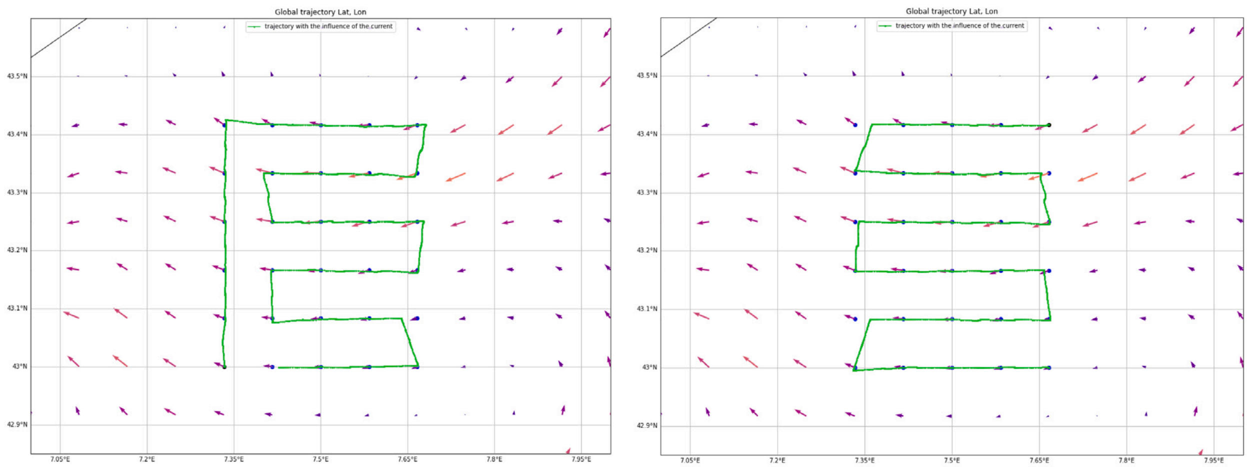
Fig. 18. Output of the SEA041 mission simulation with four more sensors, 113 dives.

Table 3
Comparison of outputs between two possible Hamiltonian paths.

Starting node	Duration of the simulated mission	Travel distance	Available energy
0	5.67 days	86.93 km	83.18%
29	5.19 days	66.64 km	84.61%

7. Conclusion & perspectives

In this article, we have presented the main principles of the development of a mission simulator dedicated to path planning for underwater gliders. It is inspired by the actual operating mode of the underwater glider SeaExplorer designed and produced by Alseamar (automatic piloting through the mission management system GLIMPSE of a manually defined mission). It is intended to be used as a mission preparation tool, and then, subsequently, for the supervision of missions. This simulator has been confronted with a real experimental mission which was carried out in January 2022. A comparison has been performed on the obtained results, using the same input data in both cases. The different output data of the simulator are fully consistent with the observed results and with the data recorded during the considered mission. They emphasize the relevance of our simulator and its validity as a mission preparation tool. Concerning the next stage i.e. mission supervision, our simulator opens up new promising perspectives due to the low computational time that is required to simulate a whole mission and to play several scenarios. A trajectories simulation was also carried out in order to achieve coverage of a predefined area between Nice and Calvi. Several path planning algorithms were developed and tested, but, in this article, only the results of a Hamiltonian path generation algorithm are presented. These path planning algorithms can be used to minimize the cost function on a problem modeled by a weighted graph. From this graph, it is possible to fill an adjacency matrix using the simulator to measure the cost between any points and their neighbors and thus the edges connecting the vertices of the considered weighted graph. The



(a) Output of the Hamiltonian path generation starting from node 0

(b) Output of the Hamiltonian path generation starting from node 29

Fig. 19. Trajectories generated by the Hamiltonian path algorithm.

great interest of our simulator is to provide various adjacency matrices with several optimization criteria (energy, time or distance) moreover with different kinds of flight profile (more or less adapted to a low/high bathymetry). So the simulator is not only a tool that helps the staff to make a decision (size of the fleet, launching location, …) but it can also automatically define the complete trajectory of mission planning.

Future works will be dedicated to the comparison between other path planning algorithms. The simulation highlights the usefulness of the simulator as a decision aid tool. Actually, it not only allows to test higher level algorithms used for trajectory generation (their outputs can be used as inputs in the simulator) but it also make it possible to validate the paths obtained at the outputs of these algorithms and to choose the best solution.

The next step of these developments will be to use this simulator in order to test different scenarios obtained thanks to trajectory planning algorithms in the context of an optimal area coverage. First, the single-vehicle case will be considered, then, we will move to the multi-vehicles case. We also intend to carry out adaptive sampling with a fleet of simulated underwater gliders. To that purpose, it will surely be necessary to incorporate simulations of sensor measurements within the simulator to allow replanning according to what the drone perceives.

CRediT authorship contribution statement

Aurélien Merci: Conceptualization, Methodology, Software, Writing – reviewing. **Cédric Anthierens:** Supervision, Writing – reviewing, Conceptualization. **Nadège Thirion-Moreau:** Supervision, Writing – reviewing, Conceptualization. **Yann Le Page:** Technical manager, Writing – reviewing, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Aurelien MERCI reports financial support was provided by the French Ministry of the Armed Forces and the Defense Innovation Agency. Yann Le PAGE reports a relationship with Alseamar that includes: employment.

Data availability

The authors do not have permission to share data.

Acknowledgments

This study was conducted as part of a Cifre-Défense PhD thesis with the company Alseamar, specialist in SeaExplorer gliders. We would like to thank them, F. Besson and O. Fommervault for the field expertise that they were able to bring to us. We also wish to thank the French Ministry of the Armed Forces and the Defense Innovation Agency for the financial support they provided to this project. We also want to thank F. Imperadori and C. Vuilmet, DGA tutors of the thesis, for their support. This study has been conducted using E.U. Copernicus Marine Service Information: https://doi.org/10.48670/moi-00016.

References

Alvarez, A., Mourre, B., 2014. Cooperation or coordination of underwater glider networks? An assessment from observing system simulation experiments in the Ligurian Sea. J. Atmos. Ocean. Technol. 31 (10), 2268–2277.

Barbier, M., Bensana, E., Doose, D., Besson, F., Le Page, Y., Leopoldoff, M., 2019. A mission management system for a fleet of gliders. pp. 1–6. http://dx.doi.org/10.1109/OCEANSE.2019.8867580.

Bellicoso, C.D., Jenelten, F., Gehring, C., Hutter, M., 2018-07. Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. IEEE Robot. Autom. Lett. 3 (3), 2261–2268.

Besson, F., de Fommervault, O., Romero, J., Barbier, M., Bensana, E., Doose, D., Leopoldof, M., Larrasoain, S., 2019. A centralized web-based platform for combined glider and satellite observation analysis. In: OCEANS 2019 - Marseille. pp. 1–5. http://dx.doi.org/10.1109/OCEANSE.2019.8867287.

Bondy, J.A., Murty, U.S.R., 1976. Graph Theory with Applications. American Elsevier, New York.

Cai, J., Zhang, F., Sun, S., Li, T., 2021. A meta-heuristic assisted underwater glider path planning method. Ocean Eng. 242, 110121.

Chen, J., Jia, B., Zhang, K., 2017. Trifocal tensor-based adaptive visual trajectory tracking control of mobile robots. IEEE Trans. Cybern. 47 (11), 3784–3798.

Cieślak, P., 2019. Stonefish: An advanced open-source simulation tool designed for marine robotics, with a ROS interface. In: OCEANS 2019 - Marseille. pp. 1–6.

Codevilla, F., Müller, M., Dosovitskiy, A., López, A.M., Koltun, V., 2017. End-to-end driving via conditional imitation learning. CoRR abs/1710.02410.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16.

Eriksen, C., Osse, T., Light, R., Wen, T., Lehman, T., Sabin, P., Ballard, J., Chiodi, A., 2001. Seaglider: A long-range autonomous underwater vehicle for oceanographic research. IEEE J. Ocean. Eng. 26, 424–436.

Fommervault, O., Besson, F., Lattes, P., 2018. SeaExplorer underwater glider: A new tool to measure water velocity. Mar. Technol..

Grande, D., Huang, L., Harris, C., Wu, P., Thomas, G., Anderlini, E., 2021. Open-source simulation of underwater gliders. http://dx.doi.org/10.23919/OCEANS44145.2021.9705852.

2002. Haversine formulas. https://www.movable-type.co.uk/scripts/latlong.html, Accessed: 2022-08-02.

Imanberdiyev, N., Fu, C., Kayacan, E., Chen, I.-M., 2016. Autonomous navigation of UAV by using real-time model-based reinforcement learning. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). pp. 1–6.

Ji, H., Hu, H., Peng, X., 2022. Multi-underwater gliders coverage path planning based on ant colony optimization. Electronics 11 (19).

Juang, C.-F., Yeh, Y.-T., 2018. Multiobjective evolution of biped robot gaits using advanced continuous ant-colony optimized recurrent neural networks. IEEE Trans. Cybern. 48 (6), 1910–1922.

Koenig, N., Howard, A., 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Vol. 3. pp. 2149–2154 vol.3. http://dx.doi.org/10.1109/IROS.2004.1389727.

Lan, W., Jin, X., Chang, X., Wang, T., Zhou, H., Tian, W., Zhou, L., 2022. Path planning for underwater gliders in time-varying ocean current using deep reinforcement learning. Ocean Eng. 262, 112226.

Lan, W., Jin, X., Wang, T., Zhou, H., 2021. Improved RRT algorithms to solve path planning of multi-glider in time-varying ocean currents. IEEE Access 9, 158098–158115. http://dx.doi.org/10.1109/ACCESS.2021.3130367.

Laurent, C., 2008. Moose (dyfamed).

Lekien, F., Mortier, L., Testor, P., 2008. Glider coordinated control and Lagrangian coherent structures. IFAC Proc. Vol. 41 (1), 125–130, 2nd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.

Leonard, N.E., Paley, D.A., Davis, R.E., Fratantoni, D.M., Lekien, F., Zhang, F., 2010. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay. J. Field Robotics 27 (6), 718–740.

Liu, X., Ma, D., Yang, M., Xia, X., Guo, P., 2022. Modified block A* path-planning method for hybrid-driven underwater gliders. IEEE J. Ocean. Eng. 47 (1), 20–31.

Ma, X., Wang, Y., Zhang, G., Yang, S., Li, S., 2022. Discrete-time formation control of multiple heterogeneous underwater gliders. Ocean Eng. 258, 111728.

Mahdoui, N., Frémont, V., Natalizio, E., 2019. Communicating Multi-UAV System for Cooperative SLAM-based Exploration. J. Intell. Robot. Syst. 98 (2), 325–343.

Manhães, M.M.M., Scherer, S.A., Voss, M., Douat, L.R., Rauschenbach, T., 2016. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE Monterey. pp. 1–8.

2015. Marine copernicus database. https://resources.marine.copernicus.eu/products, Accessed: 2022-08-02.

Meurer, W.P., Blum, J., Shipman, G., 2021. Volumetric Mapping of Methane Concentrations at the Bush Hill Hydrocarbon Seep, Gulf of Mexico. Front. Earth Sci. 9, 747.

Meyer, D., 2016. Glider technology for ocean observations: A review. Ocean Sci. Discuss. 2016, 1–26.

Michel, O., 2004. Cyberbotics ltd. Webots™: Professional mobile robot simulation. Int. J. Adv. Robot. Syst. 1 (1), 5.

Phoemsapthawee, S., Le Boulluec, M., Laurens, J.-M., Deniset, F., 2013. An underwater glider flight simulator. J. Mar. Sci. Appl. 12.

Prats, M., Pérez, J., Fernandez, J.J., Sanz, P.J., 2012. An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 2577–2582.

Rahman, M.S., Kaykobad, M., 2005. On Hamiltonian cycles and Hamiltonian paths. Inform. Process. Lett. 94 (1), 37–41.

Rath, A.K., Parhi, D.R., Das, H.C., Muni, M.K., Kumar, P.B., 2018. Analysis and use of fuzzy intelligent technique for navigation of humanoid robot in obstacle prone zone. Def. Technol. 14 (6), 677–682.

Schmittle, M., Lukina, A., Vacek, L., Das, J., Buskirk, C.P., Rees, S., Sztipanovits, J., Grosu, R., Kumar, V., 2018. Openuav: A UAV testbed for the CPS and robotics community. In: 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS). pp. 130–139. http://dx.doi.org/10.1109/ICCPS.2018.00021.

2014. Seaexplorer specifications. https://www.alseamar-alcen.com/products/underwater-glider/seaexplorer, Accessed: 2022-08-02.

Shah, S., Dey, D., Lovett, C., Kapoor, A., 2017. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. CoRR abs/1705.05065.

Sherman, J.T., Davis, R.E., Owens, W.B., Valdes, J.R., 2001. The autonomous underwater glider ''Spray''. IEEE J. Ocean. Eng. 26, 437–446.

Song, Y., Naji, S., Kaufmann, E., Loquercio, A., Scaramuzza, D., 2020. Flightmare: A flexible quadrotor simulator. In: Conference on Robot Learning.

Stommel, H., 1989. The slocum mission. Oceanography.

Tai, L., Paolo, G., Liu, M., 2017. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE Press, pp. 31–36. http://dx.doi.org/10.1109/IROS.2017.8202134.

Takaya, K., Asai, T., Kroumov, V., Smarandache, F., 2016. Simulation environment for mobile robots testing using ROS and Gazebo. In: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC). pp. 96–101.

2015. Water current model - globAL_analysIS_forecaST_pHY_001_024. https://doi.org/10.48670/moi-00016, Accessed: 2022-08-02.

Webb, D., Simonetti, P., Jones, C., 2001. SLOCUM: An underwater glider propelled by environmental energy. IEEE J. Ocean. Eng. 26, 447–452.

Wen, H., Zhou, H., Fu, J., Zhang, X., Yao, B., Lian, L., 2022. Consensus protocol based attitudes coordination control for Underwater Glider formation. Ocean Eng. 262, 112307.

Wiley, J., Sons, L., 2011. Handbook of marine craft hydrodynamics and motion control. pp. 15–44. http://dx.doi.org/10.1002/9781119994138.ch2, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119994138.ch2 URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119994138.ch2.

Winkler, A.W., Bellicoso, C.D., Hutter, M., Buchli, J., 2018. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. IEEE Robot. Autom. Lett. 3 (3), 1560–1567.

Zhang, J., Tai, L., Xiong, Y., Liu, M., Boedecker, J., Burgard, W., 2018. VR goggles for robots: Real-to-sim domain adaptation for visual control. CoRR abs/1802.00265.

Zhao, J., Wang, X., Zhang, G., Neri, F., Jiang, T., Gheorghe, M., Ipate, F., Lefticaru, R., 2015. Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots. Integr. Comput. Aided Eng. 23.