Peer-Reviewed Technical Communication

Modified Block A* Path-Planning Method for Hybrid-Driven Underwater Gliders

Xing'an Liu, Dongli Ma, Muqing Yang, Xinglu Xia, and Peixu Guo

***Abstract*—In this study, a modified Block A* algorithm is applied to the path planning of hybrid-driven underwater gliders, which combine the buoyancy-driven mode of underwater gliders and the propeller-driven mode of autonomous underwater vehicles, in the ocean environment characterized by strong currents. The objective is to predict a sequence of steering directions and the appropriate driving mode, such that vehicles can minimize their energy consumption to reach the goal region. In this work, we modify the path generation approach of the original Block A* algorithm to gain more realistic paths for hybrid-driven underwater gliders and apply an effective energy consumption model in the heap value calculation to find energy-cost optimal paths. Zermelo's function is also incorporated to select the driving mode. The performance of the presented method is then discussed by conducting simulations in three different ocean environments. The results indicate that compared to other methods, the modified Block A* method avoids local turbulent flow and gains feasible paths for hybrid-driven underwater gliders at a lower energy cost.**

***Index Terms*—Block A*, energy-cost optimal path, hybrid-driven underwater gliders (HUGs), path planning.**

Fig. 1. PETREL, an HUG designed by Tianjin University.

I. Introduction

An UNDERWATER glider (UG) is a type of autonomous underwater vehicle (AUV), which is driven by the positive or negative net buoyancy to rise or dive. By changing the ballast inside, the glider can migrate vertically and transform the vertical motion to forward propulsion by the wings, forming a saw-tooth gliding pattern. Stommel [1] first proposed this concept in 1989, and over the past few years, UGs have been widely used in oceanographic research [2]–[4]. In virtue of their use of buoyancy propulsion systems and low energy consumption design, UGs are capable of long-term deployments for up to several months [5]. Nevertheless, these vehicles travel at a relatively low speed, i.e., typically no more than 0.5 m/s, whereas the maximum speed of propeller-driven AUVs can reach over 2 m/s. The low velocity of UGs results in their vulnerability to ocean currents, the velocity of which can reach a maximum of 1.7 m/s in some regions [6]. To navigate through high-speed ocean currents directly, a new concept called a hybrid-driven underwater glider (HUG) is proposed [7] (see Fig. 1). Apart from the basic configuration of the UG, it contains a propeller at the stern section [8], making it possible to travel in AUV mode at a higher speed, reaching 1.5 m/s [9]. However, related research on this new underwater vehicle is very limited, which motivates us to take full advantage of its alternative modes in different environments.

Manuscript received April 12, 2020; revised October 20, 2020 and April 6, 2021; accepted July 4, 2021. Date of publication August 24, 2021; date of current version January 13, 2022. *(Corresponding author: Muqing Yang.)*

Associate Editor: R. Bachmayer.

The authors are with the School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China (e-mail: liuxingan@buaa.edu.cn; madongli@buaa.edu.cn; yangmuqing@buaa.edu.cn; xiaxinglu@buaa.edu.cn; buaaguopeixu@buaa.edu.cn).

Digital Object Identifier 10.1109/JOE.2021.3097414

Due to the special motion characteristics of UGs, path planning is a key issue for this type of vehicle. On the one hand, because UGs may drift from their planned trajectory when they enter an ocean environment characterized by strong currents, it is essential to ensure that the destination is approachable. On the other hand, it is also advantageous to conserve energy for long-term sailing. However, path planning for UGs is of great difficulty. In addition to the restriction due to the UGs' low speed, human control is possible only when the vehicle rises to the surface, making path planning more challenging.

To date, many path-planning search algorithms have been developed for AUVs in the ocean current field. In our view, some approaches with the most relevance include evolutionary computing, graph-based search algorithms, random exploration based methods, and some other approaches.

Evolutionary computing is a commonly used approach for this kind of problem. Alvarez *et al.* [10] developed a genetic algorithm (GA) to find the path with minimum energy cost in space-time-variable ocean environments. As an extension to multiobjective problems, a multiobjective GA was applied to AUVs for interdisciplinary coastal research [11], which considered the minimization of the total travel distance and the maximization of the number of samples. Attempting to apply heuristics to evolutionary computing, Zamuda and Sosa [12] presented an approach where differential evolution is used for UG path planning, especially when dealing with opportunistic short-term missions. The advantages of evolutionary computing are the excellent convergence and robustness characteristics; however, this method has low efficiency and depends on the selection of the initial population.

Algorithms based on the geometrical graph, which is usually a discretized grid structure of the computational area, have been widely researched as well. As a classic method for path planning in a graph, a Dijkstra algorithm including the time-variant cost function was investigated to determine the effect of the grid structure [13]. In 2005, Garau *et al.* [14] first employed the A* searching procedure, which used a heuristic function to improve the efficiency of the Dijkstra algorithm and

incorporated ocean currents with a uniform grid discretization for AUV path planning. To improve the traditional A* that discretizes the possible bearings, constant-time surfacing A* (CTS-A*) was then described, allowing all bearing angles and integrating the actual glider trajectory [15]. Based on CTS-A*, Zhou *et al.* [16] extended the method to enable an adjustable speed and employed a precise energy consumption model. Grid-based approaches tend to obtain optimal outcomes, but the difficulty is how to adapt them for a continuous space.

Since the search spaces are high dimensional, quick exploration using probabilistic sampling methods has also been researched. Rapidly exploring random trees (RRTs) [17] is an efficient means of this type and has been applied to path planning for conventional AUVs [18] and gliders [6]. Despite its robustness, this method cannot ensure optimality and performs poorly in highly constrained environments [19].

In addition to the abovementioned methods, other optimization methodologies based on level-set equations [20]–[22], iterative optimization [23], and recursive-greedy algorithms [24] have also been discussed.

Although numerous path planning algorithms have been presented for AUVs, there has been limited research addressing path planning for HUGs. Existing research mainly focused on the depth control [25] and navigation of HUGs [26]. Ramos *et al.* [27] realized real-time path planning for a hybrid Slocum glider based on the dynamical system theory, but they did not take full advantage of the two propel modes. HUGs combine the attributes of conventional propeller-driven AUVs and buoyancy-driven UGs, which means that the driving mode can be switched between these alternatives. Therefore, one of the aims of this study is to explore an approach to make the most of this characteristic in different ocean environments. In addition, as A* algorithm has the potential to find optimal paths, we attempt to increase the performance when it is used for path planning in ocean currents by introducing a modified algorithm.

In this article, we employ a database-driven search algorithm, namely, Block A*[28], to address the path planning problem for HUGs and incorporate the optimal navigation formula from Zermelo as a criterion for selecting driving modes. Compared with A*, Block A* is feasible to find shorter and more realistic paths because a precomputation procedure for possible patterns is increased during the process. Furthermore, Zermelo's equation is used to instruct which driving mode will consume less energy.

The rest of this article is organized as follows. Section II first briefly introduces Block A* algorithm and Zemerlo's theory, and then the proposed path planner is described in detail. Section III presents the numerical simulation results of the path planner compared to other methods in different ocean environments. Finally, Section IV concludes this article.

II. Methods

In this section, we will first introduce the original Block A* algorithm and the theory of Zermelo's optimal navigation formula, which are the basis of our research, and then the proposed path planning method is discussed.

A. Block A*

A* is a widely used heuristic algorithm in path planning, which considers a heuristic function as an estimation from the current position to the target to improve computational efficiency. For every grid, a *heap value*, which is the sum of the actual cost (g-value) and the estimation (h-value), is on behalf of its possibility to be a node of the optimal path.

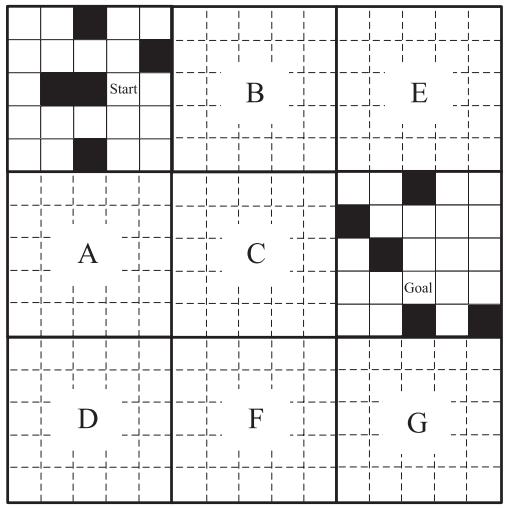


Fig. 2. Typical search map of the Block A* algorithm, where a block is composed of 5 × 5 cells.

In this way, unnecessary detection can be avoided, and the heuristic function crucially influences the outcome gained by A* [29].

To enhance the performance of A* under rigorous conditions, which include poor heuristics, dynamic maps, and real-time constraints, a modified path-planning algorithm Block A* was presented by Yap *et al.* [28]. Compared with traditional A* that manipulates a single cell at a time, Block A* copes with all cells in a block ($m \times n$ regions of cells), with the help of the local distance database (LDDB). For each boundary cell of a local block, an optimum path starting from it to every boundary cell on all four sides of the block is calculated. The actual cost of all paths is stored in the LDDB so that information regarding all possible tracks across a block is precomputed. During the process of Block A*, the *heap value* is attached to the block and can be easily updated by invoking the LDDB, so that a group of cells is explored simultaneously. In addition, because the basic path planning is conducted in a local block whose size is limited, breadth-first search is capable with acceptable computational cost, ensuring that the optimal path is found. By this means, Block A* is scarcely sensitive to the choice of the heuristic function. However, the size of the block affects the complexity of the computation of the LDDB. Supposing that each cell is either occupied or free, there are $2^{b \cdot b}$ possible patterns of grid obstructions for a $b \times b$ block of cells. This means that a feasible b value is restricted to a small value (no more than 5 according to the reference work) as a result of the exponential growth in size. However, a 2×2 block already outperforms A* by a speed factor of 2.

Here, we give a brief introduction to the Block A* algorithm and the pseudocode is shown in Algorithm 1. Similar to other grid-based methodologies, the map is first discretized into *cells*, which are further grouped into regions of $m \times n$ contiguous grid cells called *blocks*. For clarification, a typical search map of Block A* is shown in Fig. 2 (the black cells are obstacles), with a block of size 5×5 as an example. As previously mentioned, the LDDB of the current search area is then generated. The start block and goal block are initialized according to the start cell and goal cell (lines 1–2 in Algorithm 1, ① in Fig. 3). Afterward, the block with the least heap value in the *Open* list is selected as the current block (lines 4–6 in Algorithm 1), and expanded to its neighbor block to update their heap values (lines 10–21 in Algorithm 1, ② in Fig. 3). When the goal block is finally visited and there is no chance to find better solutions (lines 8–9 in Algorithm 1, ③ in Fig. 3), a path from the start cell to the goal cell will be reconstructed (lines

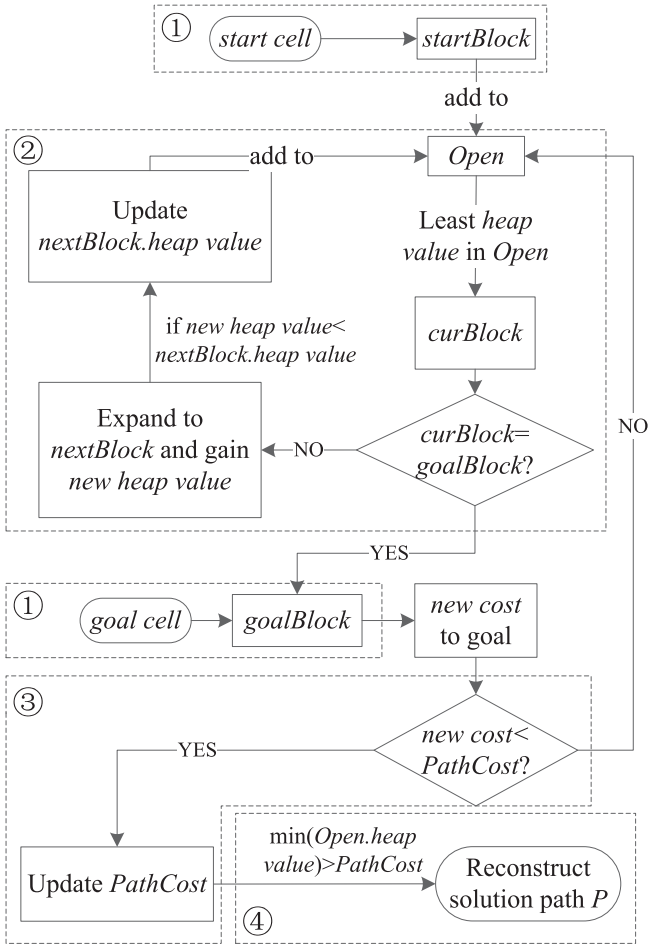


Fig. 3. Flowchart of the Block A* algorithm.

24–26 in Algorithm 1, ④ in Fig. 3). The pseudocode of this procedure is shown in Algorithm 1, and a flowchart is displayed in Fig. 3.

B. Zermelo's Optimal Navigation Formula

The criterion of selecting a proper driving mode for HUGs in our approach is inspired by Zermelo's optimal navigation formula. Zermelo's navigation problem, proposed in 1931 by Ernst Zermelo, is a classic time-optimal problem that deals with point-to-point navigation in a flow [30]. In this case, the maximum speed of the vehicle is regarded as a certain value, and the goal is to generate the optimal control to reach the destination point in the least time. After the problem was presented, it was then studied with a number of extensions and variations in other works of literature (see [31] and [32], for example). The optimal control scheme for the general case in the x–y plane is in the form of a partial differential equation, known as Zermelo's equation [30]

$$\frac{d\theta}{dt} = \frac{\partial v}{\partial x}\sin\theta^2 + \left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)\cos\theta\sin\theta - \frac{\partial u}{\partial y}\cos\theta^2 \quad (1)$$

where the speeds of the flow in the x-axis and y-axis are $u(x, y)$ and $v(x, y)$, respectively, and θ denotes the bearing angle of the boat.

Although Zermelo's optimal navigation formula only fits the path planning problem for time-continuous bearing control, while HUGs can only be controlled when intermittently floating to the surface, we can still make use of the equation for a reference. This will be discussed further in the following section.

Algorithm 1: Block A*.

Require: LDDB, start cell *start*, goal cell *goal*
Ensure: optimal path *P* from *start* to *goal*
PROC: Block A*(LDDB, *start*, *goal*) **return** *P*
1: *startBlock* = initial(*start*) ▶ determine the block of start cell and the shortest path from start to all boundary cells
2: *goalBlock* = initial(*goal*) ▶ determine the block of goal cell and the shortest path from all boundary cells to goal cell
3: *PathCost* = ∞ ▶ denote the total cost
4: insert *startBlock* into *Open*
5: **while** (*Open* ≠ empty) **and** ((*Open*.top).heap value < *PathCost*) **do**
6: *curBlock* = *Open*.top ▶ choose the current block
7: *Y* = set of all *curBlock*'s ingress cells
8: **if** *curBlock* = = *goalBlock* **then**
9: *PathCost* = $\min_{y\in Y}$(*y.g*+*cost*(*y*, *goal*), *PathCost*)
10: **else** ▶ expand to the neighbor block from current block
11: **for** every egress cell *x* of *curBlock* with neighbor *nextBlock* **do**
12: *x.g* = $\min_{y\in Y}$ (*y.g*+LDDB(*y*, *x*), *x.g*) ▶ the optimal path from the ingress cell to egress cells
13: *x'.g* = min(*x.g*+cost(*x*, *x'*), *x'.g*) ▶ get to the neighbor boundary cell *x'*
14: *new_heapvalue* = $\min_{upadated\ x'}$ (*x'.g* + *x'.h*)
15: **if** *new_heapvalue* < *nextBlock*.heapvalue **then**
16: *nextBlock*.heapvalue = *new_heapvalue*
17: insert *nextBlock* into *Open*
18: **else**
19: insert *nextBlock* into *Close*
20: **end if**
21: **end for**
22: **end if**
23: **end while**
24: **if** *Cost* ≠ ∞ **then**
25: Reconstruct solution path *P*
26: **else**
27: **return** Failure
28: **end if**

C. Path Planning Method Based on Block A* and Zermelo's Equation

This study adopts a methodology based on Block A* to find the optimal path (least energy cost) for HUGs originating from a start point to a goal point. It takes Zermelo's optimal navigation formula for the criterion to determine whether the buoyancy-driven mode or the propeller-driven mode should be used. Before the introduction of the path planner, some basic assumptions regarding the problem discussed in this article are made as follows.

1) The ocean surface is approximated to be a 2-D plane so that distances between points are calculated as a simple Euclidean distance. A spherical distance is more precise for the earth surface, but this approximation is adequate for exploring an effective path. Given the longitude and latitude of two points A (LonA, LatA) and B (LonB, LatB), which define a near-to-rectangular region, the distance in the x-direction is calculated as $|111 \times \cos(\text{LatA}/2 + \text{LatB}/2) \times (\text{LonB} - \text{LonA})|$ km, and the distance in the y-direction is calculated as $|111 \times (\text{LatB} - \text{LatA})|$ km.
2) Information on ocean currents is thought to be available and time-invariable during a glider mission. It is reasonable to assume this since the computation time of path planning is quite short

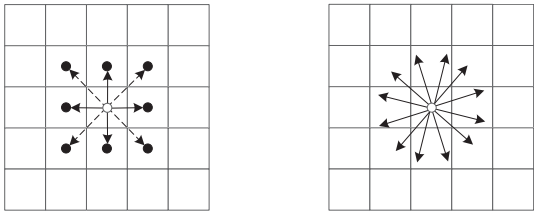


Fig. 4. Direction of every move is limited to four or eight headings in the A* algorithm (left), whereas the vehicle can actually move in all directions (right).

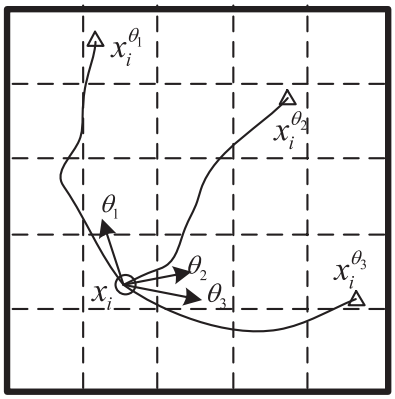


Fig. 5. Calculation for every move in the modified Block-A* method. Starting from the location x_i, considering different bearing angles θ_k and integrating the actual glider track during a cycle time.

in comparison with the period of the dynamic ocean current. The existing ocean current prediction systems provide hourly predictions [15], and for a time-variable flow, the algorithm could regenerate a new path rapidly as long as the updated ocean data are transmitted to the HUG when it surfaces. In this case, a constant ocean current is sufficient to test the efficiency of the proposed path planning method.

1) Path Generation Method: Analogous to the Block A* algorithm, a precomputed local cost database (LCDB) based on the optimal paths between every two boundary cells belonging to one block is required by our path planner. Nevertheless, graph-based search techniques, such as the A* algorithm mentioned previously, are generally deemed to generate infeasible trajectories since the tracing points are compelled to coincide with the predefined discrete grid nodes. With vehicle motion limited to four or eight headings in the original algorithms, as shown in Fig. 4, the solution space degrades and the result may fall into local minima when extended to realistic continuous space. To avoid these issues, a number of alternations are made to the original standard Block A* in regard to computing every step of the movement, gaining more realistic paths.

First, the bearing angles of the glider extend to all angles from 0° to 360° (k sample bearings selected for implementation during the computation). Then, for each angle θ_k, we choose a step time Δt to integrate the actual glider track originating from node x_i during a cycle time t_r (shown in Fig. 5). When the HUG finally reaches the end point of each track, we will attach its real location to the nearest cell according to the actual location. If more than one location is stored in the same cell, then the location closest to the goal cell will be chosen. Notably, cells are only used for judgment during the algorithm, and the real position stored in the corresponding cell will be used as the start when another cycle departs from this cell. By calculating motions in this way when

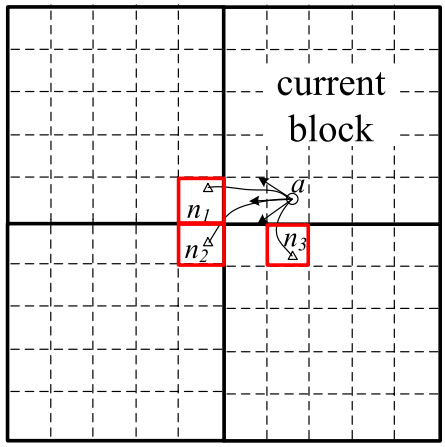


Fig. 6. Examination of the neighbor block for the current block. Cells n_1, n_2, and n_3 are all neighbor cells for cell a in the modified Block A* method.

building an LCDB or other cases, a wider solution space is set up, and the output path is more realistic and instructive.

Another notable change to the original Block A* included in our algorithm is a different definition of the neighboring blocks. In [28], the movement of each process is constrained to four headings, which makes it easy to determine a neighbor block. Nevertheless, path planning for HUGs involving ocean currents is totally different. If the vehicle moves downstream, then it can reach a block that does not adjoin to the start cell. Hence, for each block, we redefine its neighbor by testing which block could reach during a diving cycle starting from all the boundary cells of the current block. For example, as shown in Fig. 6, although geometrically, only cell n_3 is the neighbor cell of cell a, cell a can also reach cells n_1 and n_2 in a period. However, inevitably, for every ingress cell of the neighboring block, more than one boundary cell of the current block can obtain access to it. On account of this, only the cell whose real location is the nearest to the goal cell will be reserved.

2) Application of Zermelo's Equation for the Mode Choice: The choice between two driving modes in the proposed method needs to be discussed. In most cases, there are no obstacles in the sea, but some areas are virtually inaccessible for buoyancy-driven gliders because of strong ocean currents. In this situation, the HUG should switch to AUV mode for a higher speed to finish routes that are impossible for buoyancy-driven gliders. In another case, although the glider may reach its goal cell, a longer-distance trajectory is applied to avoid unreachable areas. Zermelo's equation is then employed to detect this case. For a set of start and goal points during LCDB construction or the initialization of the start block and goal block, the glider mode will primarily be used for finding an optimal path $P = \{p_1, p_2 \dots p_{n-1}, p_n\}$. In addition to the intermediate points ($p_2 \sim p_{n-1}$) stored, the determined bearing angle θ_i to get to the next process at each point is obtained as well. Then, the difference between the angles of the adjacent process ($\theta_{i-1} - \theta_i$) is computed and compared with the $\Delta\theta$ obtained by Zermelo's equation. If these two values differ significantly from each other, over 15°, for example, then it is considered necessary to test whether the propeller-driven mode is more effective. For a gained path, this usually happens in a series of neighboring points, so the propeller will be activated during the selected section as a potential alternative.

3) Energy Model: As one of the aims of this study is energy conservation, the *heap value* used in the algorithm is not the time cost but the energy cost. Consequently, a more precise energy consumption model is necessary, since a switch between two modes is taken into consideration for HUGs. An effective energy consumption model for

UGs made up of several subsystems was introduced in [33], and is verified by experiments. We employ this model to calculate the *g-value* for the glider mode and the energy consumption for the AUV mode can be easily estimated given the hydrodynamic parameters.

The energy consumption during a gliding cycle mainly contains two parts: one part is the power consumption of the buoyancy-regulating device and the pitch-regulating device, which is related to the rising and diving motion, and the other part is related to time, which is the power consumption of the sensor, control unit and other onboard devices. The buoyancy-regulating device works to change the volume at the surface and the maximum depth. Given the driving-buoyancy force ΔB, the volume change ΔV can be computed as

$$\Delta V = \frac{2\Delta B}{\rho g} \tag{2}$$

where ρ is the density of the ocean and g is the acceleration of gravity. Then, using the operational powers P_1 and P_2, as well as the flow rates q_1 and q_2, the energy consumption for the buoyancy-regulating device E_{buoyancy} can be obtained as

$$E_{\text{buoyancy}} = \frac{2\left|\Delta B\right|}{\rho g}\left(\frac{P_1}{q_1} + \frac{P_2}{q_2}\right). \tag{3}$$

The pitch-regulating device operates twice in a gliding cycle to adjust the pitch angle to the necessary angle by shifting the internal movable mass. According to [33], the energy consumption E_{pitch} can be calculated as

$$E_{\text{pitch}} = \frac{4mP_p\left|z_g \tan\left(\gamma+\alpha\right)\right|}{m_p v_p} \tag{4}$$

where v_p is the speed of the movable mass, P_p is the power of the pitch-regulating device, x_g and z_g are coordinates of the center of gravity of the vehicle, and γ and α are the gliding angle and the angle of attack, respectively.

Consequently, the energy cost of the gliding mode is the sum of E_{buoyancy}, E_{pitch}, and time-related energy consumption E_t, addressed as follows:

$$\begin{aligned} E_{\text{glide}} &= E_{\text{buoyancy}} + E_{\text{pitch}} + E_t \\ &= \frac{2\left|\Delta B\right|}{\rho g}\left(\frac{P_1}{q_1} + \frac{P_2}{q_2}\right) \\ &\quad + \frac{4mP_p\left|z_g \tan\left(\gamma+\alpha\right)\right|}{m_p v_p} + P_t t \end{aligned} \tag{5}$$

where P_t is the power of time-related devices, and t is the running time. The detailed instructions of all the symbols are listed in Table I.

In regard to AUV mode, the energy consumption is the sum of the energy cost by the propeller E_{propller}, used for overcoming the drag, and other equipment E_t. The hydrodynamic drag D for the vehicle traveling at the velocity v_{pro} in the AUV mode is modeled as

$$D = 0.5\rho v_{\text{pro}}^2 S C_{D0} \tag{6}$$

where S is the sectional area of the main body and C_{D0} is the drag coefficient. Taking into consideration the efficiency of the propeller system η_{pro}, E_{propller} can be calculated as

$$E_{\text{propller}} = D \cdot v_{\text{pro}}/\eta_{\text{pro}} \cdot t = 0.5\rho v_{\text{pro}}^3 S C_{D0} t/\eta_{\text{pro}}. \tag{7}$$

The energy consumption model for the AUV mode can then be obtained as

$$E_{\text{AUV}} = E_{\text{propller}} + E_t = 0.5\rho v_{\text{pro}}^3 S C_{D0} t/\eta_{\text{pro}} + P_t t. \tag{8}$$

Detailed explanations for the symbols are listed in Table I as well.

TABLE I
PARAMETERS OF THE HUG

Parameters	Explanations	Values	Units
m_{glider}	Weight of the HUV	125	kg
C_{D0}	Drag coefficient	0.453	-
C_{Di}	Induced drag coefficient	0.0057	-
$C_{L\alpha}$	Lift coefficient	0.5694	/°
S	Sectional area of the main body	0.053	m^2
P_p	Power of the pitch-regulating device	10	W
m_p	Movable mass	20	kg
v_p	Speed of the movable mass	1.5	mm/s
z_g	Z-coordinate of the center of gravity	5	mm
P_t	Time dependent power	4.6	W
P_1	Power of the buoyancy-regulating device in the ocean	31.77	W
P_2	Power of the buoyancy-regulating device at the surface	5	W
q_1	Flow rate in the ocean	3.75	mL/s
q_2	Flow rate at the surface	7.5	mL/s
ΔB	Driving-buoyancy force	12	N
ρ	Density of the ocean	1025	kg/m^3
η_{pro}	Efficiency of the propeller system	0.5	-

The heuristic function is similar to the energy model used in the *g-value* calculation, except that an underestimated temporal cost to reach the goal from the current location is adopted. The Euclidean distance to the goal d_{Euc} is used, and it is considered that the vehicle moves downstream with the maximum velocity of ocean current v_{oceanmax}. In this case, the time t_h used for energy consumption in the *h-value* can be expressed as

$$t_h = \frac{d_{\text{Euc}}}{\left|v_{\text{glider}}\right| + \left|v_{\text{oceanmax}}\right|}. \tag{9}$$

Although this heuristic function was employed widely in [6], [14], [15], and [16], there is not enough evidence to prove its optimality. However, the function only works when the block expands in our Block A* path planner and does not affect the LCDB, which stores the optimal path. It is reasonable that the proposed approach significantly reduces the disadvantage of an ineffective heuristic function.

4) Modified Block A Method for HUGs:* On the basis of what has been discussed, to solve the point-to-point path planning problem for HUGs, the proposed modified Block A* method is demonstrated in Algorithm 2 and operates as follows. In the first step, the LCDB is constructed for a given discretized domain, and the start block and the goal block are initialized, which means that all paths from boundary cells to the start or goal cell are generated (lines 1–2). Then, for every block in the Open list O, the block with the lowest heap value will be taken as the current block, and the start block is chosen as the first block at the beginning (lines 4–6). After that, the neighboring block of the current block is determined (lines 7–8) and the current block expands to the next (lines 9–22). This process is repeated until the goal block is visited and the *heap value* of every block in O is higher than the discovered path. The final step of the algorithm is to reconstruct the solution path from the goal cell to the start cell, similar to A*.

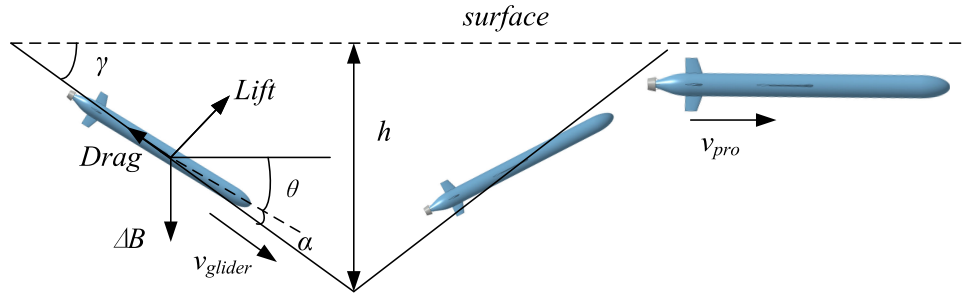


Fig. 7. Motion parameters for buoyancy-driven mode and AUV mode.

Algorithm 2: Modified Block A* method for HUGs.

Require: LCDB, start cell *start*, goal cell *goal*
Ensure: optimal path *P* from *start* to *goal*
PROC: pathplanner(LCDB, *start*, *goal*) **return** *P*
1: *startBlock* = initial(*start*) ▶ determine the block of start cell and the shortest path from start to all boundary cells
2: *goalBlock* = initial(*goal*) ▶ determine the block of goal cell and the shortest path from all boundary cells to goal cell
3: *Cost* = ∞ ▶ denote the total cost
4: insert *startBlock* into *O*
5: **while** (*O* ≠ empty) **and** ((*O*.top).heap value < *Cost*) **do**
6: *curBlock* = Open.top ▶ choose the current block
7: *X* = set of all *curBlock*'s egress cells
8: *N* = neighbor(*curBlock*) ▶ set of all *curBlock*'s neighbor blocks
9: **for** every *nBlock*∈*N*
10: *Y* = set of all *nBlock*'s ingress cells ▶ based on *X*
11: **if** *nBlock* = = *goalBlock*
12: *Cost* = $\min_{y\in Y}$(*y*.*g*+*cost*(*y*, *goal*), *Cost*)
13: **else**
14: *new_heapvalue* = $\min_{y\in Y}$ (*y*.*g*+*y*.*h*)
15: **if** *new_heapvalue* < *nBlock*.heapvalue **then**
16: nBlock.heapvalue = new_heapvalue
17: insert *nBlock* into *O*
18: **else**
19: insert *nBlock* into *C*
20: **end if**
21: **end if**
22: **end for**
23: **end while**
24: **if** *Cost* ≠ ∞ **then**
25: Reconstruct solution path *P*
26: **else**
27: **return** Failure
28: **end if**

III. Simulation and Results

A. Initial Conditions

To calculate the energy consumption in the simulation, the detailed motion parameters of the HUG are given in this section. For the buoyancy-driven mode, the maximum depth of every diving cycle is $h =$ 200 m and the gliding angle is fixed at $\gamma = 20°$. The velocity in this mode is $v_{\mathrm{glider}} = 0.5$ m/s, leading to a period of 0.65 h between each surfacing. When the propeller is on, the velocity of the glider increases to a higher velocity v_{pro}. Six samples of v_{pro} can be selected in this mode, which are

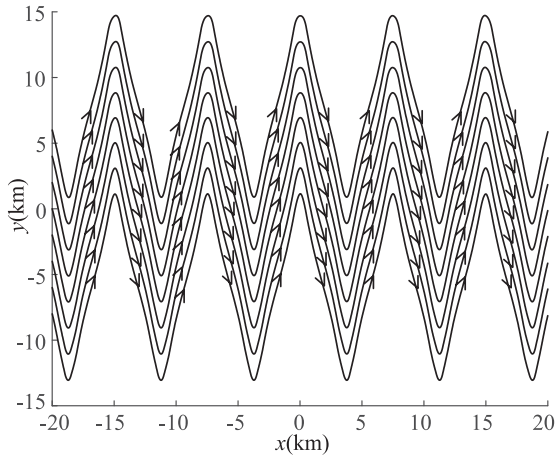


Fig. 8. Flow field of the meandering jet model.

$v_{\mathrm{pro}} = [1, 1.1, 1.2, 1.3, 1.4, 1.5]$ m/s. The hydrodynamic coefficients of the vehicle are calculated by using the computational fluid dynamics (CFD) method, which has been widely used for other UGs. The energy parameters of the buoyancy-regulating device and the pitch-regulating device are referenced to [33] and [34], and the power of the time-related devices is the sum of sensors and the control unit. In addition, the efficiency of the propeller system is determined according to the result of the experiment. All the vehicle parameters are listed in Table I, and some of the motion parameters and their relationships are shown in Fig. 7.

Simulations are conducted under the environment of MATLAB, and a 5 × 5 block is employed for the proposed Block A* approach. For comparison, a direct-to-goal (DTG) strategy, an RRT-based method, and a CTS-A* method are adapted under the same conditions. The DTG strategy is the case where the vehicle's heading is always set to the destination at every surfacing. The RRT and CTS-A* methods used in the simulation are based on the work from Rao and Williams [6] and Fernández-Perdomo *et al.* [15]. In addition, for the CTS-A* and Block A* methods, the same equivalent discretization level of 10° in the bearing grid is selected.

B. Simulation in a Meandering Jet

Since the vertical components of ocean currents are ignorable compared with the lateral currents and are inaccurately measured, the ocean current is limited in the 2-D plane in this study, with a velocity composed of $u(x, y)$ and $v(x, y)$. The ocean current model used in this section is based on a meandering jet in the eastward direction, which is a simple mathematical model of the Gulf Stream [35]. This model can

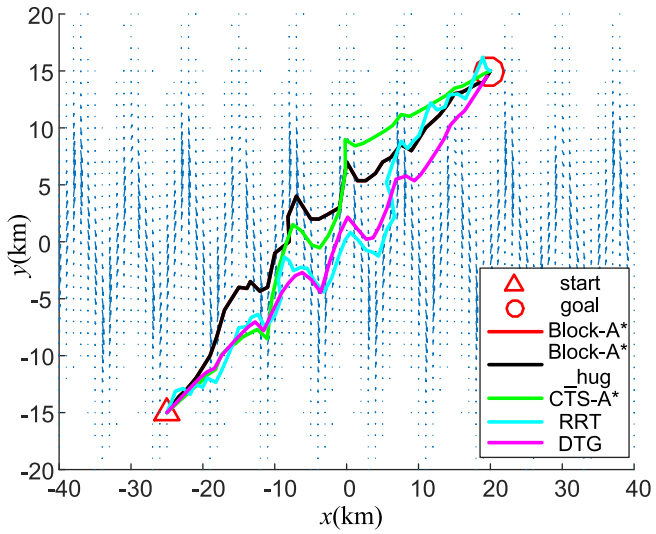


Fig. 9. Comparison of paths for mission 1 in the meandering jet. Blue arrows represent the ocean currents field.

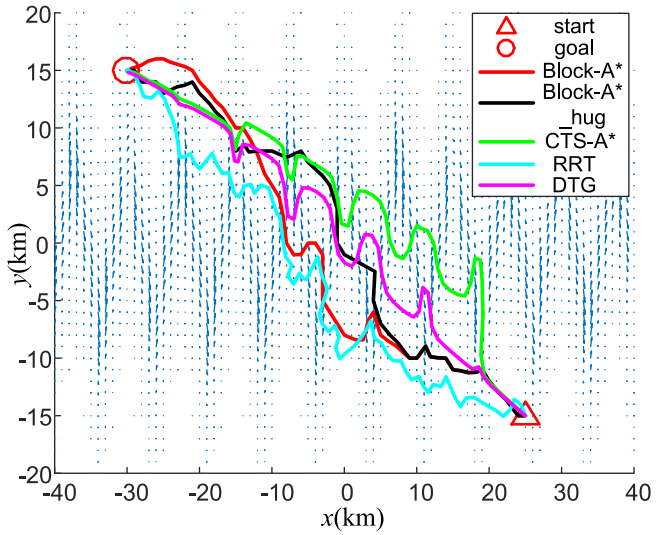


Fig. 10. Comparison of paths for mission 2 in the meandering jet. Blue arrows represent the ocean currents field.

be represented by the stream function $\varphi(x, y)$ as follows:

$$\varphi(x,y) = 1 - \tanh\left(\frac{y - B(t)\cos(k(x-ct))}{\left(1 + k^2 B(t)^2 \sin^2(k(x-ct))\right)^{0.5}}\right). \tag{10}$$

The velocity of the ocean current can be obtained as follows:

$$u(x,y,t) = -\frac{\partial\varphi}{\partial y}, v(x,y,t) = \frac{\partial\varphi}{\partial x}. \tag{11}$$

This function was applied in [10], [13], and [36] to test the path planning algorithm in a 2-D ocean environment, and it was modified to fit the problem discussed in this article. Instead of using a time-dependent meander amplitude $B(t)$, a constant $B_0 = 1.95$ is adopted for simplification, which does not affect the algorithm according to the assumption we presented previously. To create an area that is wide enough for the simulation, kilometer-scale x and y are utilized, and the ocean model is expanded by a factor of 4 on the y-axis. The other parameters are set as $k = 0.84$, the same as [13], and $t = 0$ for a time-invariant current. Hence, the final ocean model used can be described as follows:

$$\varphi(x,y) = 1 - \tanh\left(\frac{y/4 - B_0\cos kx}{\left(1 + k^2 B_0^2 \sin^2 kx\right)^{0.5}}\right). \tag{12}$$

The flow field of the applied ocean model is shown in Fig. 8. Under this condition, the maximum flow velocity of the global domain is up to 1.36 m/s, a fairly high velocity for UGs.

The first case of the simulation is quite easy for path planning, in which the track from the start point to the goal is basically downstream. As shown in Fig. 9, all of the path planners are able to perform and even the DTG method yields satisfactory results. The path gained by the CTS-A* method costs less energy than the DTG method but requires more time. Additionally, the RRT algorithm yields the worst result with rapid fluctuations. Here, we employ the Block A* method for UGs and HUGs, respectively, but they are the same under this condition. The energy cost for each approach is listed in Table II, and the Block-A* method yields a more effective path than the other methods.

TABLE II
ENERGY AND TIME COST RESULTS FOR SIMULATIONS IN THE MEANDERING JET

Methods	Mission 1		Mission 2	
	Energy Cost(J)	Time cost(h)	Energy Cost(J)	Time Cost(h)
Block-A*	8.64×10^5	25	1.22×10^6	35.1
Block-A*_hug	8.64×10^5	25	1.17×10^6	24.2
CTS-A*	9.06×10^5	27.4	1.33×10^6	41.1
DTG	9.27×10^5	26.8	1.35×10^6	38.7
RRT	1.24×10^6	35.1	1.71×10^6	48.2

TABLE III
COMPUTATIONAL COST FOR EACH METHOD

Methods	Mean computing time for path planning (seconds)	Mean computing time for the LCDB (minutes)
Block-A*	12.5	7.7
Block-A*_hug	13	16.3
CTS-A*	55	-
DTG	<5	-
RRT	10.2	-

TABLE IV
ENERGY COST RESULTS FOR DIFFERENT BLOCK SIZES

Methods	Energy Cost(J)	Mean computing time for path planning (seconds)	Mean computing time for LCDB (minutes)
5×5 Block-A*	1.01×10^6	12.5	7.7
4×4 Block-A*	1.03×10^6	23	4.7
3×3 Block-A*	1.03×10^6	37	2.4
CTS-A*	1.16×10^6	55	-

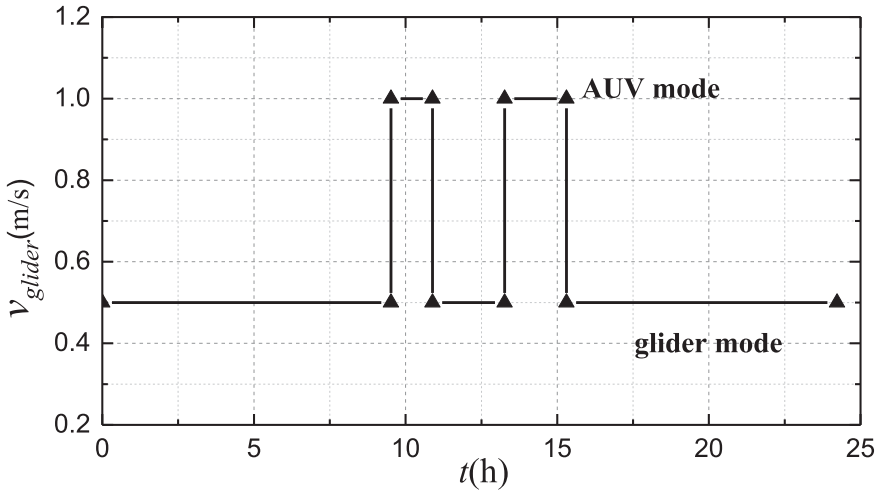


Fig. 11. Specific control scheme of the velocity for mission 2 in the meandering jet.

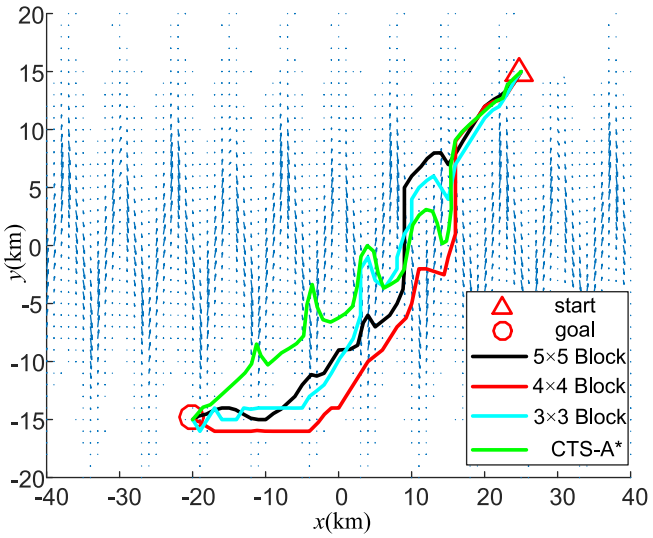


Fig. 12. Effect of the block size on the path found by the modified Block A* method. Blue arrows represent the ocean currents field.

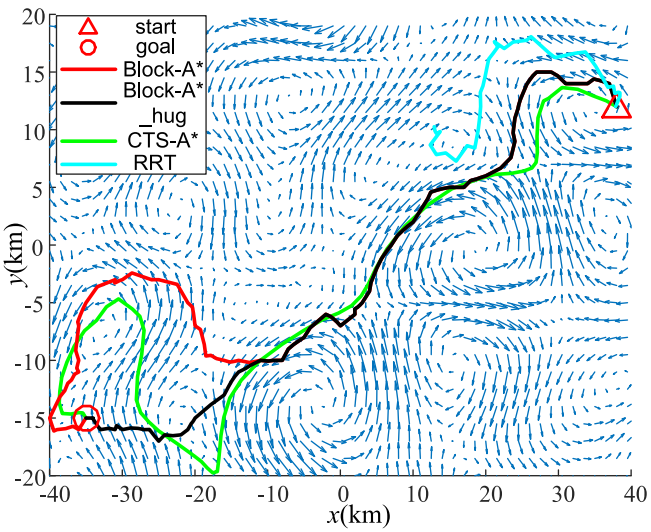


Fig. 13. Comparison of paths for mission 1 in the complex ocean environment. Blue arrows represent the ocean currents field.

TABLE V
ENERGY AND TIME COST RESULTS FOR SIMULATIONS IN THE COMPLEX ENVIRONMENT

Methods	Mission 1		Mission 2	
	Energy Cost(J)	Time cost(h)	Energy Cost(J)	Time cost(h)
Block-A*	1.26×10^6	35.7	1.35×10^6	38.1
Block-A*_hug	1.06×10^6	22.5	1.29×10^6	29.0
CTS-A*	1.35×10^6	38.1	1.48×10^6	41.6
RRT	-	-	-	-

TABLE VI
ENERGY AND TIME COST RESULTS FOR SIMULATIONS IN THE REAL ENVIRONMENT

Methods	Energy Cost(J)	Time Cost(h)
Block-A*	1.26×10^6	35.7
Block-A*_hug	1.26×10^6	35.7
CTS-A*	1.75×10^6	49.4
RRT	-	-

The ocean environment in the second mission is more complicated than that in the first task when the start point and the goal point change, as shown in Fig. 10. Although the RRT-based method obtains a feasible result, it consumes the most energy (presented in Table II) and still cannot avoid unnecessary fluctuations. The CTS-A* method exhibits a lower energy consumption than RRT and DTG, but it cannot avoid the adverse currents as much as possible. The proposed Block A* method leads to a fewer energy-cost path than other methods. Compared with the UGs, the algorithm for HUGs acquires a more direct outcome with a higher energy efficiency and less time cost. The switch between two modes during the path is shown in Fig. 11. The results show that the

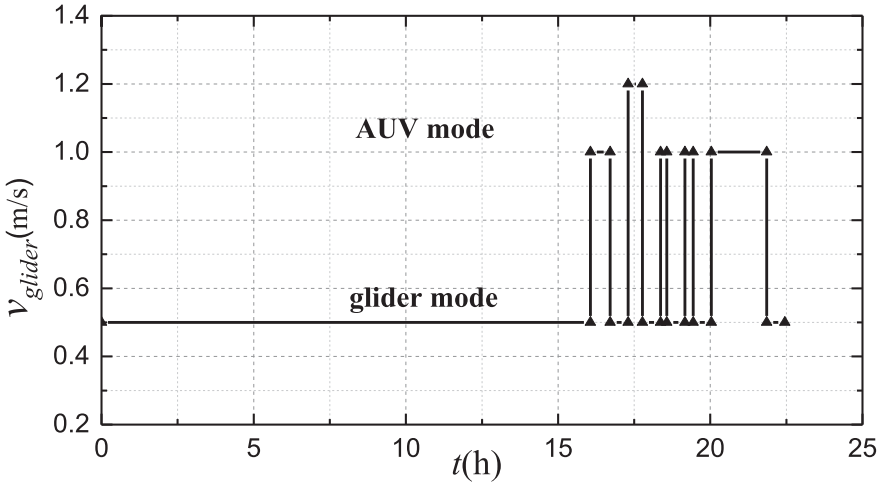


Fig. 14. Specific control scheme of the velocity for mission 1 in the complex ocean environment.

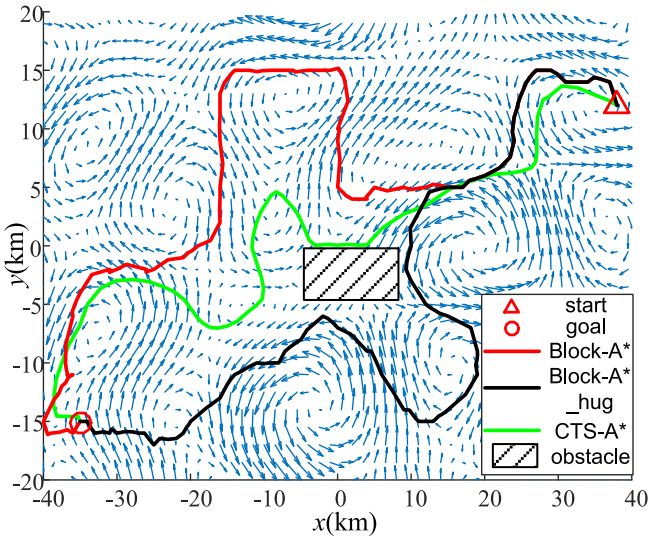


Fig. 15. Comparison of paths for mission 2 in the complex ocean environment. Blue arrows represent the ocean currents field.

method presented in this research results in up to 6.8% and 13.2% energy conservation in these two cases, which confirms the advantage of this new approach.

The computing time for each method is listed in Table III, obtained on an Intel(R) Core(TM) i5 quad processer computer running at 2.4 GHz with 8 GB of memory. The results show that the two Block-A* methods run three times faster than the CTS-A* method, and the mean computing time for path planning is only 0.6% of the glide cycle.

C. Effect of the Block Size

The effect of the block size is then investigated by adopting three different block sizes, which are 5×5, 4×4, and 3×3. With the best performance in all other methods, CTS-A* is selected for comparison. The simulation is executed in the meandering jet environment , and the result is shown in Fig. 12 and Table IV. For the Block A* method, although the outcomes change as the size of the block varies, the energy costs are similar to each other and consistently lower than that of the path obtained by CTS-A*, as they all avoid fluctuations in an adverse current. For a block with a larger size, mode judgment works better, so it gains a more straight-to-goal path. Considering the computing time, when the block size is smaller, it takes less time for the construction of LCDB but takes more time for path planning.

D. Simulation in an Ocean Environment With Complex Spatial Variability

In this section, a more complex ocean environment including several eddies is used for simulation (see Fig. 13). The stream function $\varphi(x, y)$ is obtained from an isotropic power spectrum with random phases, which has been used for generating ocean environment in [10] and [14], and the maximum velocity in the current field is set as 1.5 m/s.

As shown in Fig. 13, the RRT method could not find the path to the goal point in this complex field. Because in the region near the start point, the feasible channel passing through the complex eddies is narrow. For the RRT method, which explores the whole region randomly, it is hard to find this path in the limited computing time, which is 3 min in this simulation. The path gained by the CTS-A* method is influenced by the eddy near the goal point and is caught in the strong adverse current when it tries to reach the goal point. All these effects lead to extra cost. The modified Block A* method avoids the adverse current predictively, and takes advantage of the eddy over the goal point, leading to a 6.7% lower energy cost than the CTS-A* method (see Table V). By switching to AUV mode near the goal point (see Fig. 14), the Block A* method for the HUG reaches the goal point directly, saving 21.5% energy consumption and 13.2 h to finish the task.

Fig. 15 shows the results of the second mission when an obstacle is located to block the trajectories found in Fig. 13. The path found by the CTS-A* method is basically constituted by portions from the previous trajectory in the unblocked area plus extra portions surrounding the obstacle, and it still cannot avoid adverse currents. Two Block-A* methods, nevertheless, all lead to new paths and utilize new eddy structures. The comparison indicates that the proposed path planning method takes full advantage of the global information and therefore leads to paths with lower energy costs. Although the energy and time cost of paths found by Block-A* methods increase, they are still lower than the result of the CTS-A* method (see Table V).

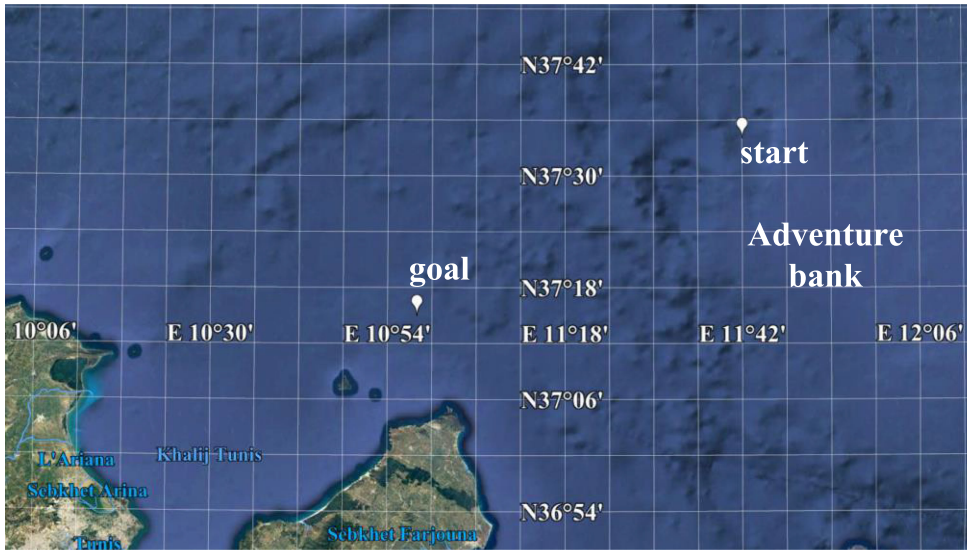


Fig. 16. Region for simulation in the Tunisia-Sardinia-Sicily region.

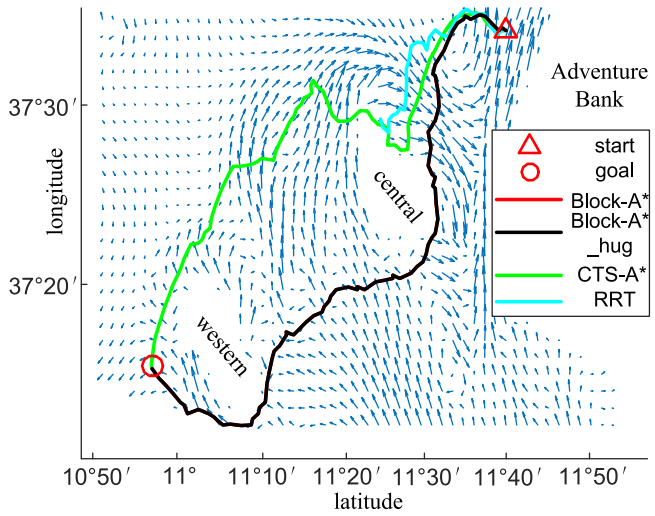


Fig. 17. Comparison of paths for simulation in the west of the Adventure bank. Blue arrows represent the ocean currents field. White regions represent the regions where the water is shallower than 200 m.

E. Simulation in a Near-to-Real Ocean Environment

The modified Block A* algorithm can now be used to find the path with minimum energy consumption in a near-to-real ocean environment. The Tunisia–Sardinia–Sicily region, located in the Mediterranean Sea, is selected for the simulation. Based on the data originating from the hydrographic survey performed in this area in 1996 [37], Onken *et al.* [38] calculated the current fields in this area. The ocean model showed the complex bathymetry and current fields in this region, and the time-averaged currents west of the Adventure bank were given with a horizontal resolution of 1.5 km. Therefore this region is chosen for simulation.

The simulation region is from N37°10′ E10°50′ to N37°40′ E12° (see Fig. 16), which is mostly deeper than 250 m and appropriate for the proposed glider motion. Since the maximum depth of every gliding cycle is 200 m, the current field used is the average of the data at depths of 100 and 150 m. The regions where the water is shallower than 200 m are regarded as inaccessible and left white in Fig. 17. As shown in Fig. 17, the RRT method fails to find the path to the goal point since it cannot bypass the central shallow water region. The CTS-A* method does not avoid the adverse currents northwest of the central region and leads to a higher energy cost path (see Table VI). Since the maximum current velocity is 0.5 m/s, which is close to the vehicle velocity in glider mode, the outcomes of the two Block-A* methods are identical. These methods take advantage of the current around the central region and western region, decreasing energy consumption by 28%.

IV. Conclusion

In this article, we present a novel path planning method based on the Block A* algorithm and employ Zermelo's optimal navigation formula to adapt it for HUGs, which can switch between AUV mode and glider mode. The original Block A* algorithm is adapted so that it can generate more realistic paths and better suit the path planning issue for HUGs. Then, Zermelo's function is applied to determine which driving mode is more suitable in the present ocean environment for HUGs. With the use of an accurate energy consumption model, the path planning method is finally completed. The effect of the proposed method was evaluated numerically by simulations and the following conclusions can be drawn.

1) The modified Block A* path planning method with Zemerlo's function can figure out the optimal path planning problem for HUGs, which determines the heading angles as well as the driving mode.
2) Contrary to the original A* and other grid-based methods, the modified Block A* method generates more realistic and informed paths.
3) Three groups of simulations were conducted in different ocean environments and the results proved the advantage of the modified Block A* method, especially in complex environments. Whether for UGs or HUGs, the Block A* method led to lower energy conservation and gained up to 28% energy conservation in contrast with the best of other alternative methods.

4) By contrasting three different block sizes in the same case, further research shows that the size of the block has little impact on the performance of Block A*.

One direction of future research is to minimize the size of the LCDB since for a long-term mission, the precomputed database of a wide ocean domain demands considerable time. This work should also be extended to 3-D time-varying ocean current fields. Although numerical simulations have demonstrated the potential of the modified Block A* path planning method, it needs to be confirmed by employing the algorithm on HUGs. In future work, an in-lake test of the proposed method will be conducted to validate the offline generated mission plan based on the available glider information. If the method is validated by experiments, then field trials at sea will be conducted.

REFERENCES

[1] H. Stommel, "The Slocum mission," *Oceanography*, vol. 2, no. 1, pp. 22–25, 1989.

[2] C. C. Eriksen *et al.*, "Seaglider: A long range autonomous underwater vehicle for oceanographic research," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 424–436, Oct. 2001.

[3] J. Sherman, R. E. Davis, W. B. Owens, and J. Valdes, "The autonomous underwater glider 'Spray'," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 437–446, Oct. 2001.

[4] D. C. Webb, P. J. Simonetti, and C. P. Jones, "SLOCUM: An underwater glider propelled by environmental energy," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 447–452, Oct. 2001.

[5] J. G. Graver, "Underwater gliders: Dynamics, control and design," Ph.D. dissertation, Dept. Mech. Aerosp. Eng., Princeton Univ., Princeton, NJ, USA, 2005.

[6] D. Rao and S. B. Williams, "Large-scale path planning for underwater gliders in ocean currents," in *Proc. Australas. Conf. Robot. Autom.*, 2009.

[7] S. X. Wang, X. J. Sun, Y. H. Wang, J. G. Wu, and X. M. Wang, "Dynamic modeling and motion simulation for a winged hybrid-driven underwater glider," *China Ocean Eng.*, vol. 25, no. 1, pp. 97–112, Apr. 2011.

[8] B. Claus, R. Bachmayer, and C. D. Williams, "Development of an auxiliary propulsion module for an autonomous underwater glider," *Proc. Inst. Mech. Eng. M, J. Eng. Maritime Environ.*, vol. 224, no. 4, pp. 255–266, Jun. 2010.

[9] F. Liu, Y. H. Wang, Z. L. WU, and S. X. Wang, "Motion analysis and trials of the deep sea hybrid underwater glider Petrel-II," *China Ocean Eng.*, vol. 31, no. 1, pp. 55–62, Jan. 2017.

[10] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.

[11] A. Moura, R. Rijo, P. Silva, and S. Crespo, "A multi-objective genetic algorithm applied to autonomous underwater vehicles for sewage outfall plume dispersion observations," *Appl. Soft Comput.*, vol. 10, no. 4, pp. 1119–1126, Sep. 2010.

[12] A. Zamuda and J. D. H. Sosa, "Differential evolution and underwater glider path planning applied to the short-term opportunistic sampling of dynamic mesoscale ocean structures," *Appl. Soft Comput.*, vol. 24, pp. 95–108, 2014.

[13] M. Eichhorn, "Optimal path planning for AUVs in time-varying ocean flows," in *Proc. 16th Symp. Unmanned Untethered Submersible Technol.*, 2009, pp. 45–55.

[14] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: An A* approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 194–198.

[15] E. Fernández-Perdomo *et al.*, "Path planning for gliders using regional ocean models: Application of *Pinzón* path planner with the ESEOAT model and the RU27 trans-Atlantic flight data," in *Proc. IEEE OCEANS Conf.*, Sydney, Australia, May 2010, pp. 1–10.

[16] Y. J. Zhou, J. C. Yu, and X. H. Wang, "Path planning method of underwater glider based on energy consumption model in current environment," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2014, vol. 8917, pp. 142–152.

[17] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, May 2001.

[18] C. S. Tan, R. Sutton, and J. Chudley, "An incremental stochastic motion planning technique for autonomous underwater vehicles," *IFAC Proc. Vol.*, vol. 37, pp. 483–488, Jul. 2004.

[19] M. Kalisiak and M. van de Panne, "RRT-blossom: RRT with a local flood-fill behavior," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 1237–1242.

[20] T. Lolla, P. F. J. Lermusiaux, M. P. Ueckermann, and P. J. Haley, "Time-optimal path planning in dynamic flows using level set equations: Theory and schemes," *Ocean Dyn.*, vol. 64, no. 10, pp. 1373–1397, Sep. 2014.

[21] T. Lolla, P. J. Haley, and P. F. J. Lermusiaux, "Time-optimal path planning in dynamic flows using level set equations: Realistic applications," *Ocean Dyn.*, vol. 64, no. 10, pp. 1399–1417, Sep. 2014.

[22] D. N. Subramani and P. F. J. Lermusiaux, "Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization," *Ocean Model.*, vol. 100, pp. 57–77, 2016.

[23] J. Isern-González, D. Hernández-Sosa, E. Fernández-Perdomo, J. Cabrera-Gámez, A. C. Domínguez-Brito, and V. Prieto-Marañón, "Path planning for underwater gliders using iterative optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1538–1543.

[24] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4791–4796.

[25] B. Claus, R. Bachmayer, and L. Cooney, "Analysis and development of a buoyancy-pitch based depth control algorithm for a hybrid underwater glider," in *Proc. IEEE/OES Auton. Underwater Veh.*, Sep. 2012, pp. 1–6.

[26] M. Zhou, R. Bachmayer, and B. deYoung, "Mapping the underside of an iceberg with a modified underwater glider," *J. Field Robot.*, vol. 36, no. 6, pp. 1102–1117, 2019.

[27] A. G. Ramos *et al.*, "Lagrangian coherent structure assisted path planning for transoceanic autonomous underwater vehicle missions," *Sci. Rep.*, vol. 8, pp. 1–9, 2018.

[28] P. Yap, N. Burch, R. C. Holte, and J. Schaeffer, "Block A*: Database-driven search with applications in any-angle path-planning," in *Proc. 25th AAAI Conf. Artif. Intell.*, Jan. 2011, pp. 120–125.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[30] E. Zermelo, "Über das Navigationsproblem bei Ruhender oder Veränderlicher Windverteilung," *ZAMM J. Angew. Math. Mech.*, vol. 11, no. 2, pp. 114–124, 1931.

[31] B. Li, C. Xu, K. L. Teo, and J. Chu, "Time optimal Zermelo's navigation problem with moving and fixed obstacles," *Appl. Math. Comput.*, vol. 224, pp. 866–875, 2013.

[32] G. C. Hays, A. Christensen, S. Fossette, G. Schofield, J. Talbot, and P. Mariani, "Route optimisation and solving Zermelo's navigation problem during long distance migration in cross flows," *Ecol. Lett.*, vol. 17, no. 2, pp. 137–143, Feb. 2014.

[33] J. C. Yu, F. M. Zhang, A. Q. Zhang, W. M. Jin, and Y. Tian, "Motion parameter optimization and sensor scheduling for the sea-wing underwater glider," *IEEE J. Ocean. Eng.*, vol. 38, no. 2, pp. 243–254, Apr. 2013.

[34] S. X. Wang, Y. Song, Y. H. Wang, W. D. Niu, and Z. Lei, "Sensitivity analysis of energy consumption parameters for Petrel-II underwater glider based on Sobol' method," *J. Tianjin Univ. (Sci. Technol.)*, vol. 50, no. 2, pp. 113–120, 2016.

[35] M. Cencini, G. Lacorata, A. Vulpiani, and E. Zambianchi, "Mixing in a meandering jet: A Markovian app," *J. Phys. Oceanogr.*, vol. 29, no. 10, pp. 2578–2594, Oct. 1999.

[36] M. Eichhorn, "Solutions for practice-oriented requirements for optimal path planning for the AUV 'SLOCUM Glider,'" in *Proc. MTS/IEEE OCEANS Conf.*, Sydney, Australia, Sep. 2010, pp. 1–10.

[37] R. Onken and J. Sellschopp, "Water masses and circulation between the eastern Algerian basin and the Strait of Sicily in October 1996," *Oceanol. Acta*, vol. 24, no. 2, pp. 151–166, Apr. 2001.

[38] R. Onken, A. R. Robinson, P. F. J. Lermusiaux, P. J. Haley, and L. A. Anderson, "Data-driven simulations of synoptic circulation and transports in the Tunisia-Sardinia-Sicily region," *J. Geophys. Res.*, vol. 108, no. 9, pp. 8123–8136, Sep. 2003.

Xing'an Liu received the B.E. degree in aircraft design and engineering, from Beihang University, Beijing, China, in 2016, where he is currently working toward the Ph.D. degree with the School of Aeronautic Science and Engineering.

His research interests include new concept underwater vehicle design, path planning for underwater vehicles, and dynamics of underwater vehicles.

Dongli Ma received the B.E. and M.E. degrees in fight dynamics and the Ph.D. degree in aircraft design from Beihang University, Beijing, China, in 1987, 1990, and 1996, respectively.

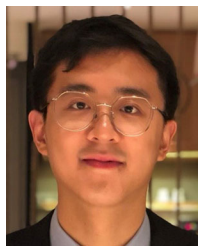
From 1998 to 1999, he was the Visiting Scholar with the Department of Aeronautical Engineering, Moscow Aviation Institute, Moscow, Russia. He is currently a Professor with the School of Aeronautic Science and Engineering, Beihang University. His research interests include aircraft conceptual design and unmanned vehicles design.

Muqing Yang received the B.E. degree in reliability system engineering and the Ph.D. degree in aircraft design from Beihang University, Beijing, China, in 2008 and 2014, respectively.

He is currently a Lecturer with the School of Aeronautic Science and Engineering, Beihang University. His research interests include new concept aircraft design and aerodynamic optimization of aircrafts.

Xinglu Xia received the B.E. degree in aircraft design, from Beihang University, Beijing, China, in 2016, where he is currently working toward the Ph.D. degree with the School of Aeronautic Science and Engineering.

His research interests include new concept aircraft design, eVTOL aircraft design and control, etc.



Peixu Guo received the B.E. degree in aircraft design and engineering, from Beihang University, Beijing, China, in 2016, where he is currently working toward the Doctoral degree in fluid mechanics.

His research interests include computational fluid dynamics, flow control, adjoint-based shape optimization and path planning of vehicles, boundary layer stability, and sensitivity analysis. He has authored or coauthored several papers in his areas of interest.