IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 14, 2020, accepted August 27, 2020, date of publication August 31, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020629

Multi-AUV Collaborative Operation Based on Time-Varying Navigation Map and Dynamic Grid Model

XIWEN MA[1], CHEN YANLI[1], GUIQIANG BAI[1], AND JUN LIU[2], (Student Member, IEEE)
[1]School of Mechanical and Aerospace Engineering, Jilin University, Changchun 130022, China
[2]School of Electronics and Information Engineering, Beihang University, Beijing 100191, China

Corresponding author: Chen Yanli (chenyanli@jlu.edu.cn)

This work was supported in part by the Jilin Province Key Science and Technology Research and Development Project under Grant 20180201040GX, in part by the National Natural Science Foundation of China under Grant 51505174, in part by the Scientific and Technological Development Program of Jilin Province of China under Grant 20170101206JC, in part by the Foundation of Education Bureau of Jilin Province under Grant JJKH20170789KJ, in part by the Research Fund for the Doctoral Program of Higher Education of China under Grant 20130061120038, in part by the National Key Research and Development Program of China under Grant 2017YFC0602002, and in part by the Aeronautical Science Foundation of China under Grant 2019ZA0R4001.

ABSTRACT In a dynamic and complex environment, to improve the cooperative operation efficiency of multiple AUV groups, a bionic neural wave network (BNWN) algorithm, and a velocity vector synthesis (VVS) algorithm are proposed. A strategy of space decomposition and node space recursion is adopted to provide dynamic navigation maps for AUV monomers and to modularize the tasks. A closed boundary function is introduced to construct a dynamic grid model to autonomously avoid obstacles with multiple moving forms. The results of three sets of simulation experiments show that the number of changes in direction, the total path length, and the collision rate of AUV individuals are greatly reduced. These results prove that the proposed algorithm has high autonomy and strong adaptability.

INDEX TERMS Boundary function, multiple AUV, space decomposition, dynamic navigation maps, dynamic grid model.

I. INTRODUCTION

Due to the shortcomings of low efficiency, narrow coverage, and insufficient research and development capabilities, AUV monomers, and therefore, multi-AUV underwater operations, have become the focus of research [1]. For multi-AUV system underwater collaborative operation, the most important thing is to be able to complete the task efficiently, accurately, and steadily with limited energy support. In the process of multi-AUV system cooperation, AUV individuals can detect the environment through their sensors and process the information returned by each sensor to achieve the purpose of estimating and determining the positions of obstacles and targets [2]; then, combined with the control algorithm and path planning algorithm, the target can be captured. However, in complex environments, it is difficult to meet the requirements of accurate, efficient, and stable operation only through sensor data processing and the fusion of traditional algorithms [3]. In addition, because, the time delay error and integral cumulative error of a sensor are difficult to eliminate, research on control algorithms has gradually become a research hotspot. After years of research and exploration, many control algorithms have emerged. Currently, widely used path planning algorithms include ant colony algorithms [4]–[6], bee swarm algorithms [7], [8], the virtual artificial potential field method [9], [10], quasi-annealing algorithms [11], Neural network algorithms [12]–[14] and particle swarm optimization [15]–[17]. However, the most commonly used task allocation strategies are artificial self-organizing neural network algorithms (SOM) [18] and tree structure algorithms [19]. To improve the adaptive ability of AUVs, Huang *et al.* [20] proposed a cooperative search path planning algorithm for multiple underwater robots based on biologically inspired neural networks. However, the grid division method is simple, and unchanged so it has difficulty meeting the requirements of high

The associate editor coordinating the review of this manuscript and approving it for publication was Weiguo Xia.

efficiency, stability, and accuracy in a dynamic environment. Cheng *et al.* [21], proposed a full coverage path planning algorithm based on graph theory knowledge. This algorithm decomposes the space into multiple subspaces to adapt to multiple forms of a robot and realizes the optimization of the path. Although the research integrates graph theory knowledge and multiple decompositions of space, this decomposition process is an offline decomposition, which obviously cannot adapt to a complex dynamic environment. Luo and Yang [22], proposed an AUV exploration algorithm based on a bionic neural network. The algorithm can be applied to an unknown environment with obstacles, and through full coverage, search to obtain the required information to build an environment model. Although this algorithm is suitable for complex dynamic environments, the algorithm has high complexity and large robot energy consumption and is not suitable for long-term online planning. Cao and Zhu [23] proposed a multi-AUV path planning and task assignment method based on BISOM and VS algorithms. However, in this method, an AUV cannot achieve variable-step autonomous obstacle avoidance, the influence of dynamic obstacles is not considered, and the adaptive ability is weak. Guo and Zhang [24], employed graph theory knowledge to apply a curvature constraint to a path planning algorithm, update the path planning algorithm, and redefine polygonal obstacles by introducing influence range parameters. However, in an actual situation, the data provided by the sensor after real-time measurement of the obstacle are discrete or incomplete and have poor accuracy. The adaptive ability of the algorithm in a dynamic complex environment is not strong. Cao *et al.* [25], proposed a potential field-layered reinforcement learning method that can control multiple underwater robot systems to effectively capture a target. However, this algorithm can be improved in the processing of environmental information and the control of robot step size.

Base on the above research, combined with knowledge of graph theory, this paper proposes the concepts of spatial decomposition and sub-decomposition. The obstacle domain is divided by stimulating cell proliferation to achieve spatial decomposition. According to the results of the spatial recursion of the subgoals, the decomposed subspaces are meshed to realize the spatial sub-decomposition. Dynamic grid space is formed by the boundary function constraint and the introduction of horizontal and vertical density functions. Considering the turning radius of the robot, the energy consumption of the operation, and the interference of ocean currents, a target tracking and capturing algorithm based on a bionic neural wave network (BNWN) and a velocity vector synthesis (VVS) algorithm are proposed. The fusion of the two algorithms improves the adaptive ability of the multi-AUV system to the dynamic environment and meets the design requirements of high efficiency and mixed online and offline planning. The contribution of this research and the advantages of the proposed algorithm are as follows:

1) Cell proliferation is simulated to decompose the obstacle group and realize spatial reconstruction.
2) Solving sub-target nodes using spatial recursion strategy, and the task, which reduces the algorithm complexity.
3) Through the design of boundary functions and the introduction of horizontal and vertical grid density functions, a dynamic grid space is constructed.
4) Through the research and design of the BNWN algorithm and VVS algorithm, the adaptive ability of multiple AUV systems under a dynamic environment is improved.

The logical structure and research process of the full text are shown in the following flowchart:

In addition, compared with traditional algorithms, the algorithm and strategy proposed in this paper have the advantages of high efficiency, strong adaptive ability and low power consumption. For example, compared with the A star algorithm [26], it has the advantages of smooth trajectory, speed stability, low calculation complexity, and real-time dynamic characteristics; compared with the improved APF algorithm [27], it has the advantages of stable trajectory, fast convergence, short time consumption, high calculation accuracy and autonomous learning; compared with GBSOM and SOM [13], it has the advantages of dynamic and flexible grid space, efficient arrangement of neurons, step size can be adaptively changed, and adapt to time-varying ocean currents.

II. PROBLEM STATEMENT

The cooperative operation of multiple underwater agents requires highly efficient and accurate control algorithms. However, the complex environment, the variability of obstacle movement patterns, and the interference of ocean currents; greatly affect the efficiency of cooperative operation. Therefore, to reduce the calculation difficulty and improve the planning efficiency, this study uses basic knowledge of graph theory and spatial planning mechanisms to construct a static grid. To improve the obstacle avoidance efficiency and self-adaptability, this study introduces a boundary function and grid density function to construct a dynamic grid. To realize the utilization of and overcome ocean currents, this study proposes a fusion algorithm of a BNWN algorithm and VVS algorithm.

III. REGIONAL DIVISION AND SPATIAL DECOMPOSITION

A. DIVISION OF OBSTACLE SPACE

In the process of constructing the underwater environment, AUV individuals not only need a variety of sensors to collect external data but also need efficient and stable algorithms to process the data. The focus of the research in this section is on determining how to combine incorporate graph theory knowledge and integrate the region division strategy to efficiently process the obtained external data.

Due to the high complexity of 3-D space, the description process is cumbersome and the composition does not clearly express the content of the study, so this article uses

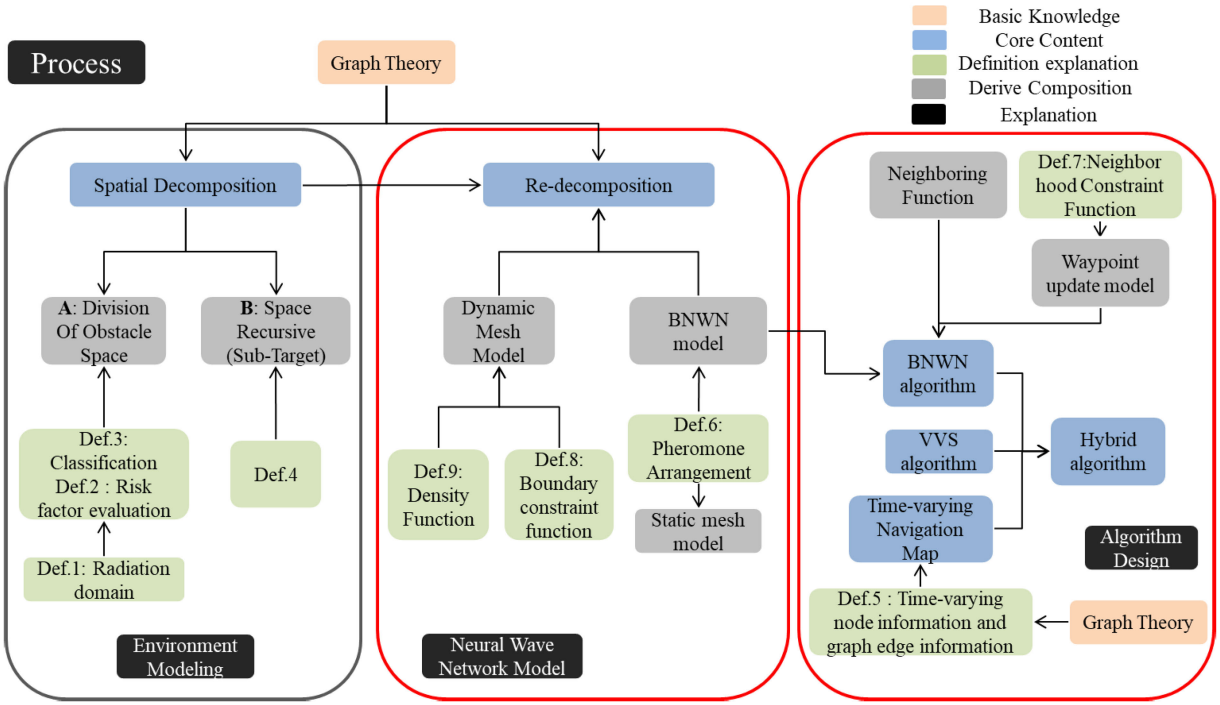

FIGURE 1. Flow chart of the full text research process.

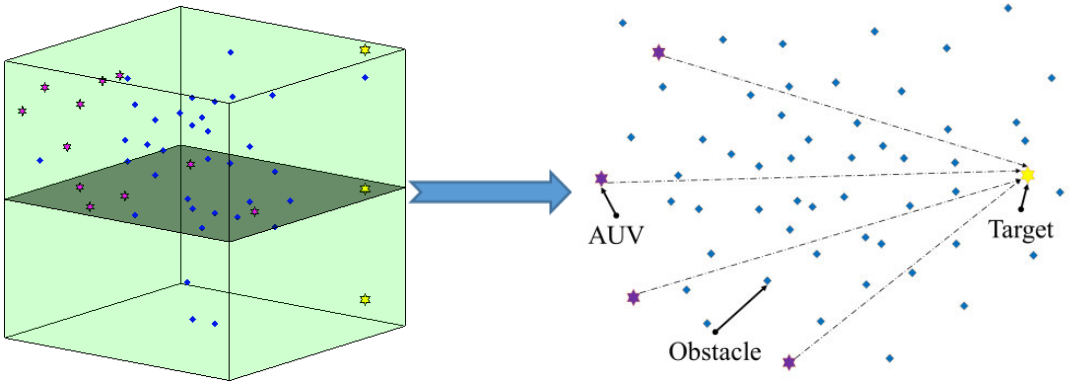

FIGURE 2. Space model diagram.

a 2-D space model for study and explanation. As shown in Fig. 2, the multi-AUV system will capture the target autonomously, collaboratively, and without collision in a complex environment.

Definition 1: The coordinate of the static obstacle center in space is $\{p_i\,(x_i, y_i)\,, i \in N^*, i < N_1\}$, and the influence range of the obstacle is expressed as a circular domain with p_i as the center and ζR as the radius.

In the definition, ζ is the hazard coefficient, R is the turning radius, and the parameter N_1 is used to control the number of obstacles. The selection of the hazard coefficient ζ is as follows:

$$\begin{cases} 0 < \zeta < 1 & if \ \left(D_{p_i} \gg 3 * R\right) \cap \left(D_{p_{i^*j}} > 2 * R\right) \\ \zeta \geq 1 & if \ \left(D_{p_i} < 2 * R\right) \cup \left(D_{p_{i^*j}} \leq 2 * R\right) \end{cases} \quad (1)$$

Figure.3 shows that each circle corresponds to an obstacle, and the influence range of the circle represents the magnitude of the risk factor. An AUV can evaluate the risk factor of each obstacle according to definition 1.

Definition 2: Obstacle hazard coefficient: $\left\{D_{p_i}, i \in N^*, i < N_1\right\}$, which is the similarity of the obstacle to the straight line L_{as}.

In the definition, the straight line L_{as} is the connection between the AUV and the target.

After the evaluation of the hazard coefficient of each obstacle is completed, the obstacles are classified and the corresponding node information is assigned to the obstacles. The node information includes: node coordinate values and node types, as shown in definition 3.

Definition 3: The obstacle that satisfies the condition $D_{p_i} < 2 * R \cap (i \in N^*, i < N_1)$ is set as a black

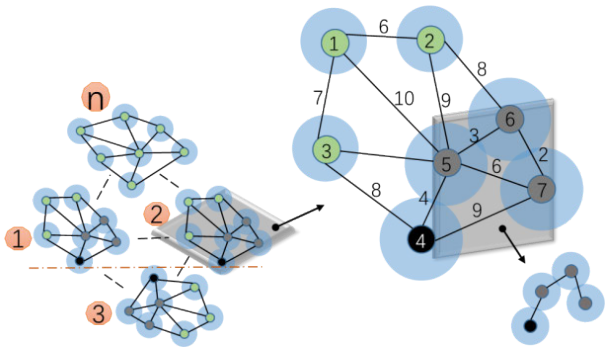

FIGURE 3. Schematic diagram of obstacle division.

node, and the black node is used as the center for local diffusion to form an obstacle domain. In the hierarchical interval $\{x * R < D_{p_i} < (x+1) * R \cap (x \in N^*, x \geq 2)\}$ of the domain space, $\{i^* = i \; if \{D_{p_{i^*j}} \leq 2 * R \cup 0 \leq 2 * abs(\sin\langle P_{ij}, P_{ij^*}\rangle) \leq \frac{1}{2+2*Q_p}\}\}$ represents the effective obstacles contained in the obstacle domain of the Q_p layer, which are represented by gray nodes. All other invalid obstacles are indicated by green nodes.

According to a graph theory theorem, the information stored in the black nodes and the gray nodes is expressed as: $\left\{D_{p_i}, p_i(x_i, y_i), D_{p_{i^*j}}\right\}$, where $D_{p_{i^*j}}$ is the similarity between the affected child obstacle and the parent obstacle in the parent obstacle domain; and the information contained in the green nodes is $\{D_{p_i}, p_i(x_i, y_i)\}$.

As shown in Fig. 4, the obstacles are divided into multiple domain spaces according to definitions 2 and 3, and then a derived subgraph with threatening obstacles is extracted according to the export subgraph lemma.

The data points of the derived subgraph are subjected to cubic Hermite interpolation, and the interpolation result is used as the boundary function of the obstacle domain for the spatial recursion of the sub-target node coordinates. The cubic Hermite interpolation formula is as follows:

$$H_3(x) = f_0\alpha_0(x) + f_1\alpha_1(x) + f_0'\beta_0(x) + f_1'\beta_1(x) \quad (2)$$

The interpolation basis function $\{\alpha_i(x), \beta_i(x)\, i = 1, 2\}$ can be expressed by the following formula:

$$\begin{aligned}\beta_0(x) &= (x - x_0)\left(\frac{x - x_1}{x_0 - x_1}\right)^2 \beta_1(x) \\ &= (x - x_1)\left(\frac{x - x_0}{x_1 - x_0}\right)^2 \end{aligned} \quad (3)$$

$$\begin{cases}\alpha_0(x) = \left(1 + 2\dfrac{x - x_0}{x_1 - x_0}\right)\left(\dfrac{x - x_1}{x_0 - x_1}\right)^2 \\ \alpha_1(x) = \left(1 + 2\dfrac{x - x_1}{x_0 - x_1}\right)\left(\dfrac{x - x_0}{x_1 - x_0}\right)^2\end{cases} \quad (4)$$

where, $\{f_i\, i = 0, 1\}$ is the function value of nodes 0 and 1, $\{f_i'\, i = 0, 1\}$ is the first derivative value of nodes 0 and 1, and $H_3(x)$ represents the coordinate value of the cubic Hermite interpolation at time t or step t or the fitted curve of a static obstacle.

The results of the obstacle division obtained by imitating the principle of cell proliferation are shown in Fig. 4. Each obstacle field contains a unique black node, which is distributed near the line between the AUV and the target. Multiple gray nodes are distributed around the black node. However, obstacles with a low-risk factor are included in the green space.

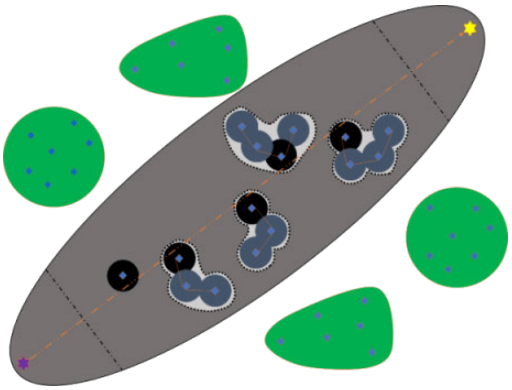

FIGURE 4. Demonstration of obstacle division.

The advantages of dividing obstacles as described above in a complex environment are as follows: First, this approach can be combined with acoustic sensors for long-distance environment modeling, which provides stable and efficient operation information for AUV underwater operations. Second, combined with the BNWN algorithm, the approach can reduce the amount of calculation, shorten the total job path length, and reduce energy consumption. Finally, combined with the graph theory theorem to store the information of graph edges and graph nodes, this approach can realize the tracking and capturing of objects with memory, and provide a shared space for the cooperative operation of multiple AUV systems.

B. SPACE RECURSIVE SOLUTION OF SUBTARGET POINT COORDINATES

To reduce energy consumption, reduce the total path length, and reduce the amount of calculation, this paper designs subtarget points near the obstacle domain based on the above research.

Definition 4: Set the threat distance of the obstacle domain to the subtarget point as $\{D = (\zeta + 1) * R, \zeta \geq 1\}$, where $\left\{R_i^n, (i = 1\,(AUV), 2\,(Sub - goal)), n \in N^*\right\}$ is the radius of the virtual diffusion domain, θ_i^n represents the diffusion angle, and α_i is the diffusion constraint angle, which is used to avoid collision and adjust the number of red nodes.

As the diffusion domain expands, θ_i^n gradually decreases, and the similarity between the intersections $A_n(x', y')$ and $T_n(x', y')$ gradually increases. After multiple iterations, the red node coordinates can be obtained as follows (5), as shown at the bottom of the next page, where, the parameter

$\left\{\{\beta_i, (i = 1, 2)\} \cap 0 < \beta_i \leq \frac{\pi}{4}\right\}$ is the gain of the diffusion constraint angle. By adjusting this parameter, the number of subgoals can be adaptively reduced during operation. The parameter ε is the discriminant factor, where $R > \varepsilon \geq 0$. Adjusting the parameter ε can be used to realize the normalization of subtarget points near the obstacle domain. The parameter K is the effective distance gain, which depends on the result of cubic Hermite interpolation and has a limit $K \geq 5$. By adjusting this parameter, the total path length can be shortened efficiently and collision avoidance for large obstacles can be achieved. The spatial recursion process of the subtarget node position is shown in Fig. 5.

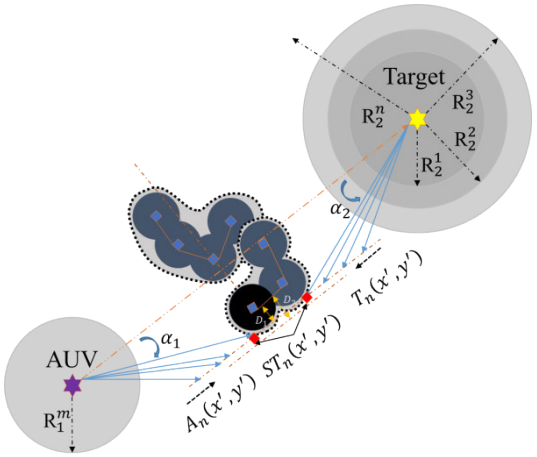

FIGURE 5. Schematic diagram of space recursion.

The spatial recursion method of the virtual sub-target node should depend on the specific working conditions. For example, when the black nodes are prominent, the following simple derivation algorithm should be used, and when the black nodes are surrounded by gray nodes, the derivation strategy mentioned in Definition 4 should be used. The following will take the example where the black node can be observed by both AUV and target. The specific solution process is as follows.

Quote: The normal vector solution formula is as follows:

$$\begin{cases} \vec{n} = \overrightarrow{P_{i-1}P_i} \times \overrightarrow{P_iP_{i+1}} = \begin{vmatrix} i & j & k \\ x_{i-1,i} & y_{i-1,i} & z_{i-1,i} \\ x_{i,i+1} & y_{i,i+1} & z_{i,i+1} \end{vmatrix} \\ \begin{cases} \overrightarrow{P_{i-1}P_i} = (x_{i-1,i}, y_{i-1,i}, z_{i-1,i}) \\ \overrightarrow{P_iP_{i+1}} = (x_{i,i+1}, y_{i,i+1}, z_{i,i+1}) \end{cases} \end{cases} \tag{6}$$

where, P_{i-1} is the current position coordinate of AUV, P_i is the coordinate of the transition sub-target position, P_{i+1} is the coordinate of the target point, and $\vec{n}$ is the normal vector of the above three points.

The virtual sub-goals are selected according to the following constraints.

1) The pheromone content of neurons around the virtual sub-target point is positive.
2) $$\begin{cases} \min\left(COS\left\langle\overrightarrow{P_iP_{i-1}}, \overrightarrow{P_iP_{i+1}}\right\rangle\right) D_{bi+1} \leq \sqrt{2}\beta\zeta \\ \qquad R \& D_{bi-1} \leq \sqrt{2}\beta\zeta R \\ \max\left(COS\left\langle\overrightarrow{P_iP_{i-1}}, \overrightarrow{P_iP_{i+1}}\right\rangle\right) D_{bi+1} > \sqrt{2}\beta\zeta \\ \qquad R \& D_{bi-1} > \sqrt{2}\beta\zeta R \\ i = 1, 2, 3, \cdots\cdots \end{cases}$$
3) $\max\left(COS\left\langle\overrightarrow{n}, \overrightarrow{n_1}\right\rangle \overrightarrow{n_1} = (0, 0, 1)\right)$
4) $0 < \arccos\left\langle\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_{i-1}P_b}\right\rangle < \arccos\left\langle\overrightarrow{P_iP_b}, \overrightarrow{P_iP_{i+1}}\right\rangle < 90$
5) $0 < \arccos\left\langle\overrightarrow{P_iP_{i-1}}, \overrightarrow{P_iP_{i+1}}\right\rangle < 90$

where, D_{bi+1} is the relative distance from the obstacle to the target point;D_{bi-1} is the relative distance from the obstacle to the AUV; β is the deformity rate or non-circular rate of the obstacle; ζR is the radius of influence of the obstacle.

Definition 5: The graph node of a subgoal is represented as $\{ST_w, w \in N^*\}$; and which is represented by a red node in the graph, and the information stored in this node is $\left\{ST_w\left(x', y'\right), \vartheta_w, a_w, \varphi_w, \lambda_{hw}, \lambda_{lw}\right\}$. Let $\left\{E_{ij}\left(G\right) i, j \in V_w\left(G\right)\right\}$ be the edge of the graph connecting the red nodes in the graph, and the storage information is $\left\{Z_{ij}, \mu_{ij}, \vartheta_{ij}\right\}$.

In the *w_th* subspace, ϑ_w is the grid vector angle, which is generally set to $0 \leq |\vartheta_w| \leq \pi$; a_w is the semi-real axis length of the grid space; and φ_w is the boundary function of the grid space, λ_{hw} is the horizontal density of the grid space, λ_{lw} is the vertical density of the grid space; Z_{ij} is the relative distance between two adjacent nodes, μ_{ij} is the number of nerve wave diffusion layers, and ϑ_{ij} is the vector angle of the boundary function.

When the subtarget position coincides with the target position, the red node position is updated to the target position coordinates. According to the various color graph nodes and graph edge constraints defined above, the 2-D space is decomposed, and the resulting topology graph is shown in Fig. 6.

The above research combines graph theory knowledge with space decomposition theory. Through the calculation of nodes, space recursion and interval planning of obstacles,

$$when\ \left\{\theta_1^m \leq \alpha_1 \cap \theta_2^n \leq \alpha_2\right\} \Rightarrow \begin{cases} ST_w\left(x', y'\right) = T_n\left(x', y'\right) orA_n\left(x', y'\right) & if\ \left\{D_{T_nA_n} < \varepsilon\right\} \\ \begin{cases} \zeta = \zeta \pm \Delta\zeta \\ \alpha_i = \alpha_i \pm \beta_i \\ ST_w\left(x', y'\right) = T_n\left(x', y'\right) orA_n\left(x', y'\right) \end{cases} & if\ \left\{\varepsilon < D_{T_nA_n} < K * R\right\} \\ ST_w\left(x', y'\right) = T_n\left(x', y'\right)\ and\ A_n\left(x', y'\right) & if\ \left\{D_{T_nA_n} \geq K * R\right\} \end{cases} \tag{5}$$

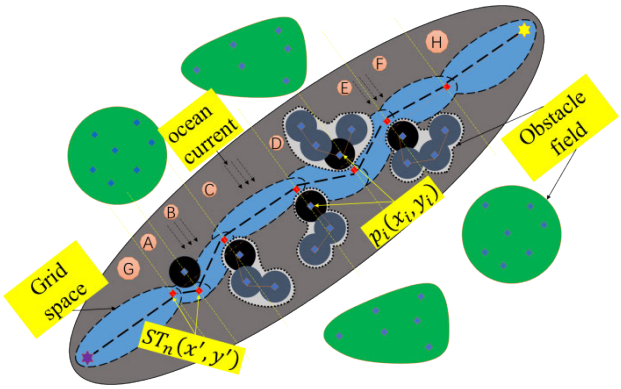

FIGURE 6. Topological diagram of environmental decomposition.

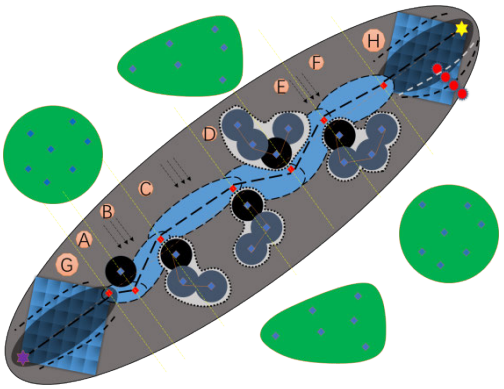

FIGURE 7. Re-decomposition diagram.

the complex environment is spatially decomposed. In addition, the task is modularized into multiple sub-tasks, and the subtasks are studied and designed below.

IV. CAPTURE ALGORITHM AND SPACE RE-CDECOMPOSITION

A. BNWN ALGORITHM

This study proposes a path planning algorithm based on the structure of a bionic neural wave network to complete subtasks. The principle of the algorithm is as follows: the subspace after the decomposition is turned into a point set, and then the point set model is virtualized into a multilayer crossing neural wave network. As shown in Fig. 7, each node in the network structure represents a neuron, and targets, AUVs, and obstacles correspond one-to-one with the virtual network structure points. Targets, obstacles, and AUVs are the source points of neural waves, where target objects and AUVs are global source points; however, obstacles are set as local source points.

Definition 6: The neurons contained in the neural wave network structure will be selectively activated by AUV individuals, the neuron pheromone content decreases layer by layer with the origin of the nerve wave as the center, the number of nerve wave layers is planned online by the boundary function, and grid density, subtarget point and AUV position.

The formula of neuron pheromone content is expressed as follows:

$$X_i(j) = E_a - \alpha_{D_{ij}} \cdot D_{ij} + \left(\frac{1}{D_{bj}} - \frac{1}{D_{\max}}\right) \cdot \frac{d_{D_{bj}}}{D_{bj}^2} \cdot \frac{\partial D_{bj}}{\partial N_b} \cdot E_b \tag{7}$$

where, $X_i(j)$ is the pheromone content of the j_th neuron in the structure of the i_th neural wave network; E_a is the pheromone content of the red node; D_{ij} is the relative distance between the red node and the neuron; D_{bj} is the relative distance between the black and gray nodes and the neuron; $D_{\max}$ is the maximum range of the local influence of the obstacle space; E_b is the pheromone content of the black and gray nodes; $d_{D_{bj}}$ stands for the local risk factor, namely, the repulsion factor; $\alpha_{D_{ij}}$ stands for the learning efficiency, that is, the gravitation factor; and $\alpha_{D_{ij}}$'s learning field is $0 < \alpha_{D_{ij}} < 1$.

The gravity factor $\alpha_{D_{ij}}$ is set using the following improved logistic function:

$$\alpha_{D_{ij}} = 1/\left(3 + \exp^{-x}\right) = 1/\left(3 + e^{-D_{ij}}\right) \tag{8}$$

The repulsion factor $d_{D_{bj}}$ can be expressed by the following formula:

$$d_{D_{bj}} = \left| \frac{1}{e^{-D_{bj}}} * \frac{2 - e^{-D_{bj}}}{1 + e^{-D_{bj}}} \right| \tag{9}$$

The formula for calculating the relative distance between nodes is as follows:

$$D_{ij} = |N_i - N_j| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{10}$$

The formula for updating the waypoints proposed in this paper is as follows:

$$R_o(t+1) = \begin{cases} R_o(t) + y_i \times [Z_n - R_o(t)], & D_{ri} > D_{\min} \\ T_l, & D_{ri} \le D_{\min} \end{cases} \tag{11}$$

where, $R_o(t+1)$ represents the navigation coordinates of the next moment of an AUV during operation; $R_o(t)$ represents the current coordinate value of the AUV; $D_{\min}$ is the shortest distance to judge whether the AUV reaches the target point; T_l is the position coordinate of the target point (red node); y_i is the domain function of the AUV; and Z_n is the next coordinate of AUV when tracking and capturing the target. When $D_{ri} > D_{\min}$, the multiple AUV system keeps tracking; otherwise, capture is performed.

The definition of the neighboring function is as follows:

$$y_i = \begin{cases} \dfrac{1}{1 + e^{-\sqrt{\left(\lambda_{hw}^2 + \lambda_{lw}^2\right)}}} & if\ D_{rj} \le D_{n\max} \\ 0 & if\ D_{rj} > D_{n\max} \end{cases} \tag{12}$$

where, λ_{hw} is the horizontal density of the grid space, λ_{lw} is the vertical density of the grid space, D_{rj}. is the relative distance between neuron and AUV.

Definition 7: The neighborhood constraint function in real-time path planning is η. By adjusting η, AUV individuals can independently choose the best waypoint to activate as follows:

$$\eta = \begin{cases} \boldsymbol{D_{rj}^{H}} < \boldsymbol{D_{n\max}^{H}} = \sqrt{\mu} * \dfrac{\mathbf{2} * \boldsymbol{a_w}}{\lambda_{\boldsymbol{hw}}} \\ \boldsymbol{D_{rj}^{L}} < \boldsymbol{D_{n\max}^{L}} = \sqrt{\mu} * \dfrac{\mathbf{2} * \left((\boldsymbol{a_w})^{\mathbf{2}} - (\boldsymbol{D_{rt}})^{\mathbf{2}}\right)^{\frac{1}{2}}}{\lambda_{\boldsymbol{lw}}} \end{cases} \quad (13)$$

where, $\boldsymbol{D_{rj}^{H}}$ is a horizontal constraint, $\boldsymbol{D_{rj}^{L}}$ is a longitudinal constraint, $\boldsymbol{a_w}$ is the semi-major axis of the boundary function, the parameter μ is the dimension control coefficient.

The specific waypoint update process is as follows:

First, calculate the neighboring neurons that meet the requirements of $\boldsymbol{D_{rj}} < \boldsymbol{D_{\max}}$, and extract the corresponding neuron coordinate values, and extract the corresponding neuron coordinate value.

Second, calculate the vector difference $\overrightarrow{\boldsymbol{V_{rj}}}$ between the AUV and neuron and the vector difference $\overrightarrow{\boldsymbol{V_{rt}}}$ between the AUV and red node, and then obtain the Cosine of vectors $\overrightarrow{\boldsymbol{V_{rj}}}$ and $\overrightarrow{\boldsymbol{V_{rt}}}$ according to the cosine theorem, that is, the similarity, as follows:

$$\begin{cases} \cos\left(\overrightarrow{\boldsymbol{V_{rt}}}, \overrightarrow{\boldsymbol{V_{rj}}}\right) = \left[\overrightarrow{\boldsymbol{V_{rt}}} \times \overrightarrow{\boldsymbol{V_{rj}}}\right] \Big/ [\boldsymbol{D_{rt}} \times \eta] \\ \overrightarrow{\boldsymbol{V_{rt}}} = \boldsymbol{N_r} - \boldsymbol{N_t} \\ \overrightarrow{\boldsymbol{V_{rj}}} = \boldsymbol{N_r} - \boldsymbol{N_j} \end{cases} \quad (14)$$

Third, calculate the difference between the similarity value and the grid vector angle value $\vartheta_{\boldsymbol{w}}$, and record it as a candidate set χ_i.

Finally, when the neuron pheromone content is positive, the next coordinate Z_n of the AUV is

$$Z_n = \left\{ \begin{array}{l} \left[\left(\min D_{tj} \cap \min\left(\chi_i\right)\right) \cup \max X_i\left(j\right)\right] \\ Z_m \in \mathring{U}\left(\varepsilon_l \varphi_w, Z_o\right) \end{array} \right\} \quad (15)$$

where, Z_o is the AUV coordinate point, Z_m is the coordinate of the *m_th* neuron in the Z_o domain, φ_w is the boundary function, and the parameter ε_l is the zoom ratio, which realizes the scaling of the boundary function φ_w.

The above research has realized the re-decomposition of space, through the selective activation of neurons under time-varying conditions to optimize the trajectory path, and finally reach the subtarget node. Multiple iterations can complete the job task.

B. BOUNDARY FUNCTION

The concept of the boundary function proposed in this paper is to enable an AUV to autonomously activate the neurons in the neural network structure, which greatly reduces the amount of calculation. AUV individuals can adjust the vector angle of the boundary function to achieve the optimization of the path point update, and can adjust the grid density to achieve the AUV variable step size of travel. The boundary function selected in this paper is the elliptic function in the conic curve.

The following is the derivation process and parameter design of the ellipse formula:

Turn the general equation of an ellipse into the following equation:

$$A\left(x - x_0\right)^2 + B\left(x - x_0\right)\left(y - y_0\right) + C\left(y - y_0\right)^2 + f = 0 \quad (16)$$

Select the parameter $\left[x', y'\right]$, and set: $x' = x - x_0 y' = y - y_0$, Then the simplified equation is as follows:

$$Ax'^2 + Bx'y' + Cy'^2 + f = 0 \quad (17)$$

For the coordinate rotation of the elliptic curve, set its rotation angle to $\{\theta_n, n \subseteq N^*\}$; then, the equation is:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta_n \sin\theta_n \\ -\sin\theta_n \cos\theta_n \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} \Leftrightarrow \begin{cases} x = x'\cos\theta_n + y'\sin\theta_n \\ y = -x'\sin\theta_n + y'\cos\theta_n \end{cases} \quad (18)$$

Substitute the result of the rotation into the standard equation and simplify it to obtion the following equation:

$$\left(a^2\sin^2\theta_n + b^2\cos^2\theta_n\right) \cdot x'^2 + \left(a^2\cos^2\theta_n + b^2\sin^2\theta_n\right) \cdot y'^2 + 2\left(a^2 - b^2\right)\sin\theta_n\cos\theta_n \cdot x'y' - a^2b^2 = 0 \quad (19)$$

After extracting the parameters and performing substitution assignment, the following formula is obtained:

$$\begin{cases} A = a^2\sin^2\theta_n + b^2\cos^2\theta_n \\ B = 2\left(-a^2 + b^2\right)\sin\theta_n\cos\theta_n \\ C = a^2\cos^2\theta_n + b^2\sin^2\theta_n \\ f = -a^2b^2 \end{cases} \quad (20)$$

Definition 8: The boundary constraint function of the neural wave network structure is $\{\varphi_w, n \subseteq N^*\}$, The grid vector angle parameter ϑ_w is equivalent to the rotation angle $\{\vartheta_n, n \subseteq N^*\}$ of the elliptic curve, The horizontal density parameter λ_{hw} of the grid space corresponds to $\frac{a_w}{\tau}$, and the vertical density λ_{lw} of the grid space corresponds to $\frac{b_w}{\kappa}$.

Definition 9: $\tau = E_1 + (-1)^{f+1} * \psi_1 * v_b$, $\kappa = E_2 + (-1)^f * \psi_2 * v_b$ are the horizontal and vertical steps of an AUV, respectively, and $T_1 < (\tau \cup \kappa) < T_2$; that is, considering the actual situation to prevent the AUV from jumping or overloading, the step constraint parameter $\{T_i i = 1, 2\}$ is proposed.

where: $\{E_i i = 1, 2E_2 \leq E_1\}$ is the set initial step constant of the AUV individual. v_b is the obstacle speed parameter, where $v_b = 0$ stands for a stationary obstacle; otherwise it represents a dynamic obstacle or other AUV individuals. $\{\psi_i i = 1, 2\}$ stands for the step learning efficiency, which satisfies $0 < \psi_i < 1$. The introduced decision parameter f has the following characteristics:

$$f = \begin{cases} 0 & if\ v_b > v_e \\ 1 & if\ 0 < v_b \leq v_e \end{cases} \quad (21)$$

where, the constant v_e is the velocity threat boundary coefficient. When the speed of an obstacle is greater than v_e, the AUV traveling at the rated maximum step at this time cannot guarantee the complete success of collision avoidance. Therefore, increasing the lateral density and decreasing the longitudinal density reduces the axial velocity, which realizes lateral azimuth collision avoidance. Conversely, reducing the lateral density and increasing the longitudinal density can achieve longitudinal acceleration and lateral deceleration, which achieves high-speed linear collision avoidance.

The change process of the neural network structure caused by the horizontal and vertical density changes in the dynamic grid space is shown in Fig. 8, where the red pentagon represents a dynamic obstacle.

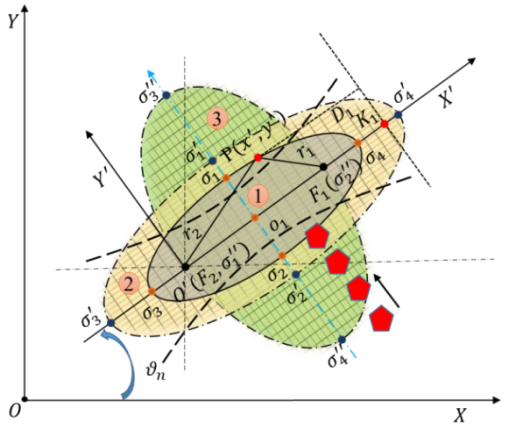

FIGURE 8. Dynamic grid space.

Figure. 8 demonstrates that boundary curve 1 represents the situation when the obstacle is stationary or there is no dynamic obstacle intrusion. Boundary curve 2 represents the situation when the moving speed of the dynamic obstacle is slow. Boundary curve 3 represents the situation when the dynamic obstacle moves faster.

An efficient and accurate recognition mechanism is the premise of a multi-AUV system to avoid dynamic obstacles autonomously. Based on this premise, this paper proposes a dangerous judgment (DJ) algorithm.

In the process of cross-variation of various working conditions, AUV individuals can collect external information through their sensors. For example, an AUV individual uses sonar to collect the coordinate information of the obstacle trajectory path points that appear in its detection S-angle. These data are used as the input of the path planning subalgorithm in the BNWN algorithm and DJ algorithm for reprocessing. Finally, the obstacles can be avoided by adjusting the grid density and the vector angle of the boundary function.

According to the real-time update characteristics of the neural network structure and the need for adaptive grid division, the important parameters contained in the boundary curve are defined as follows:

Half focal length: $c_w = \frac{D_{A_n ST_w}}{2}$;

Dangerous Judgment Algorithm for Dynamic Obstacles

Definition

OB_move: Repository of obstacle track points.

JD_AT: The intersection of the trajectory of the AUV and the obstacle

V_OB: Speed of obstacles

D_tance: Maximum sensing range of AUV monomer

Main loop:

Lck = size (OB_move,2);

If lck < 2

  Dangerous_single = 0;

Else

  Calculate:

  1. Fit the travel path of dynamic obstacles and AUV, and record them as curves **A** and **B**, respectively.

  2. Calculate the intersection point **JD_AT**;

  3. Calculate the relative distance **D_d1** of point JD_AT and AUV;

  4. Calculate the relative distance **D_d0** of point JD_AT and sub-target node;

  5. Calculate the time **Time_1** from AUV to point JD_AT;

  6. Calculate the distance **D_d00** from AUV to the sub-target node;

  7. Calculate the distance set **D_d2** of all dynamic obstacles in the library and the point JD_AT, the distance set **D_d7** from the AUV and the distance set **D_d6** from the sub-target node;

  8. [~, **d1**] = find(D_d2 == min(D_d2));

  9**. Time_2** = floor(D_d2(d1)/V_ OB);

  10**. D_tance** = D_d00* tan(S_angle);

  Decision loop:

  If D_d2(lck) < D_d2(lck−1) & D_d6 (lck−1) > D_d6(lck) & D_d7(lck−1) > D_d7(lck) & D_d 0 <= D_d00

    if D_tance< D_d7(lck) & abs(Time_2−Time_1) > 2

      Dangerous_single = 0;

    Elseif D_d2(d1) > sqrt(3*F^2)

      Dangerous_single = 0;

    Else

      Dangerous_single = 1;

    End

  End

End

Until Multiple AUV systems complete underwater tasks

Semi-short axis:

$$b_w = \begin{cases} \sqrt{a_w^2 - \left(\frac{D_{A_n ST_w}}{2}\right)^2} & if\ (f = 0 \cup v_b > v_e) \\ \frac{D_{A_n ST_w}}{2} & if\ (f = 1 \cup 0 < v_b \le v_e) \end{cases} \quad (22)$$

Semi-major axis:

$$\left\{\left(\frac{D_{A_nST_w}}{2}+\beta_1*\tau\right)\le a_w\le\left(\frac{D_{A_nST_w}}{2}+\beta_2*\tau\right)\beta_1<\beta_2\right\} \tag{23}$$

where, the role of parameter $\{\beta_i i=1,2\}$ is to reserve grid space for the reverse movement of the AUV, which meets the circumferential movement requirements of the AUV. The range is set to $3\le\beta_1<\beta_2$ according to experience. According to the above parameter definition and formula derivation, the horizontal and vertical density functions of the dynamic grid can be obtained as follows (24) and (25), as shown at the bottom of the page.

Through the above research and design, we have completed the decomposition and re-decomposition of the environment and completed the construction of the dynamic grid. Research shows that by adjusting important parameters of the boundary function, autonomous collision avoidance for static and dynamic obstacles can be achieved, and the total path length can be minimized. This paper verifies the practicability of the strategies and algorithms proposed in this paper through simulation experiments.

V. VELOCITY VECTOR SYNTHESIS ALGORITHM

Under the influence of ocean currents, the trajectory of the multi-AUV system will deviate from the shortest path planned by the BNWN algorithm. When performing tasks such as map construction, target search, tracking, and capturing, maintaining the set travel trajectory with multiple AUV systems is the key to improving work efficiency and reducing energy consumption. Therefore, it is very important to control the actual trajectory of an AUV to be consistent with the trajectory calculated by the BNWN algorithm, which is also the research focus of this chapter.

This chapter decomposes and synthesizes the speeds of AUVs and ocean currents, and proposes a VVS algorithm. In the 3-D environment, the decomposition process of AUV and ocean current velocities can be drawn from Fig.9, where vector P_x is the projection of vector OP_i on the XOY plane, and α_2 is the internal angle; vector P_y is the projection of vector OP_i on the ZOY plane, and α_2 is the internal angle; and vector P_z is the projection of vector OP_i on the ZOX plane, and α_1 is the internal angle. From this Figure, the speed decomposition expression can be derived as follows:

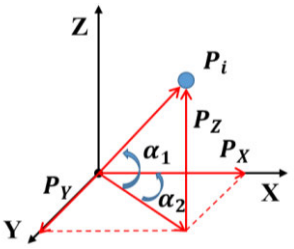

FIGURE 9. Velocity vector decomposition.

$$\begin{cases}P_x=V_0*\cos(\alpha_1)*\cos(\alpha_2)\\P_y=V_0*\cos(\alpha_1)*\sin(\alpha_2)\\P_z=V_0*\sin(\alpha_1)\end{cases} \tag{26}$$

where, V_0 is the initial setting value of the AUV velocity.

Figure. 10 shows how the VVS algorithm can optimize the waypoints of the AUV update under ocean current interference. In the Figure, the vector V_a represents the velocity of the AUV, the magnitude is adjusted according to the design parameters of the AUV, and the vector V_c represents the velocity of the ocean current. The vector V_{as} is a vector synthesis of V_a and V_c, and the three vectors are all in the same plane.

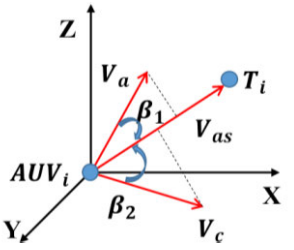

FIGURE 10. VVS algorithm implementation process.

To clearly express the VVS algorithm, Fig. 10 is converted into a 2-D plane containing vectors V_c, V_a and V_{as}, as shown in Fig. 11. In this Figure, V_{cn} is the component of ocean current velocity perpendicular to the desired trajectory direction,

$$3\le\beta_1<\beta_2\Leftrightarrow\begin{cases}\beta_1+\dfrac{D_{A_nST_w}}{2*\left(E_1+(-1)^f*\psi_1*v_b\right)}\le\left(\dfrac{a_w}{\tau}=\lambda_{hw}\right)\\\left(\dfrac{a_w}{\tau}=\lambda_{hw}\right)\le\left(\beta_2+\dfrac{D_{A_nST_w}}{2*\left(E_1+(-1)^f*\psi_1*v_b\right)}\right)\end{cases} \tag{24}$$

$$\lambda_{lw}=\frac{b_w}{\kappa}=\begin{cases}\dfrac{\sqrt{a_w^2-\left(\dfrac{D_{A_nST_w}}{2}\right)^2}}{E_2+(-1)^{f+1}*\psi_2*v_b} & if\ \ (f=0\cup v_b>v_e)\\\dfrac{\dfrac{D_{A_nST_w}}{2}}{E_2+(-1)^{f+1}*\psi_2*v_b} & if\ \ (f=1\cup 0<v_b\le v_e)\end{cases} \tag{25}$$

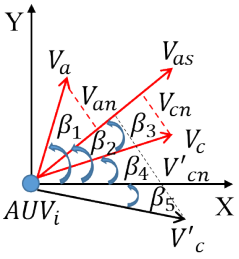

FIGURE 11. 2-D space velocity decomposition and synthesis.

and V_{an} is the component of AUV velocity perpendicular to the desired trajectory direction. In addition, V_{cl} is the component of ocean current velocity parallel to the desired trajectory direction, and V_{al} is the component of AUV velocity parallel to the desired trajectory direction.

In addition, according to the vector inside accumulate theorem, another expression is concluded as follows:

$$\beta = \arccos\left(\frac{(V_1 * V_2)}{|V_1| * |V_2|}\right) \tag{27}$$

In the formula, β is the angle between the vectors V_1 and V_2. When the vectors V_1 and V_2 represent V_a and V_{as}, β represents β_1; when the vectors V_1 and V_2 represent V_c and V_{as}, β represents β_3.

However, the size of the angle β_3 depends on the quadrant where the vector V_c is located. As shown in Fig.11, when V_c and V_{as} are in different quadrants, the angle between V'_c and V_{as} is the sum of β_2 and β_5; when V_c and V_{as} are in the same quadrant, the angle between V_c and V_{as} is the difference between β_2 and β_4.

Keeping the vertical component offset and the vector sum of the parallel components consistent with the desired trajectory direction is the key to controlling the AUV travel trajectory.

The objective function expression of the VVS algorithm can be obtained as follows:

$$\begin{cases} |V_a| * \sin(\beta_1) = |V_c| * \sin(\beta_3) \\ 0 \le \arccos\left(\dfrac{(V_a + V_c) * V_{as}}{|(V_a + V_c)| * |V_{as}|}\right) \\ |V_a| * \cos(\beta_1) + |V_c| * \cos(\beta_3) > 0 \end{cases} \tag{28}$$

Based on the above derivation, we can calculate the AUV trajectory travel angle and the desired velocity V_{as}.

The multi-AUV system can overcome the interference of ocean currents when using the VVS algorithm. In this chapter, the parameterized ocean current information (including the speed and direction of the ocean current) clearly shows the state of the ocean current. In addition, the premise of the application of the VVS algorithm is that an AUV can obtain ocean current information in real-time through its sensors (e.g., using anemometer measurements). The current information is reprocessed by the VVS algorithm, and the reprocessing result is used as the input of the bottom control system of the AUV. Sequential circulation can realize real-time control of AUV speed. However, the development of the underlying control system and how the AUV collects data through its sensors are not the focus of this article.

VI. SIMULATION ANALYSIS

There are 4 capture points evenly distributed around the target. In the simulation experiment, a simple match is made between the AUV group and the capture point according to the similarity. When all 4 capture points are occupied, it represents the completion of underwater operations.

The behavior of each monomer in the multi-AUV system during the simulation experiment is as follows: First, the environment simulation map is built independently, and the simulation map is converted into a set of points. Each point corresponds to a neuron, and the pheromone content of all neurons is set to zero. Second, based on the graph node information and the real-time update characteristics of the boundary function, the neurons in the neural wave layer are automatically selected for activation. Finally, the monomer based on the BNWN algorithm, and the VVS algorithm achieves collision-free capture of the target. During the simulation experiment, the size of the working space of the multi-AUV system is set to 300 × 300. The parameters of the simulation experiment are shown in Table 1.

TABLE 1. Control parameters.

Parameters	E_b	E_a	D_{min}	D_{max}	T_1	T_2	R
Value	-20	20	3	$\sqrt{75}$	30	30	10
Parameters	φ_1	φ_2	E_2	E_2	V_e	ε_l	S-angle
Value	0.2	0.1	5	5	8	1	$\pi/4$

A. WITHOUT CURRENT INTERFERENCE

This experiment simulates cooperative operation in low-level sea conditions, which is suitable for fault point inspection and dangerous water detection. The operation process of multiple AUV systems is simulated to track and capture the target under the condition of random distribution of various state obstacles. The functions of obstacle region division, subtarget node space recursion, and optimal path point selection in the BNWN algorithm are verified. The simulation results are shown in Fig. 12.

Figure. 12 shows the specific presentation of the calculation process of the BNWN algorithm during the collaborative operation of the multi-AUV system. In this Figure, the closed elliptic curve is used to represent the update of the boundary function in real-time. The curve marked in green indicates the boundary constraint of neural wave diffusion when the AUV is traveling normally. However, the black marked curve indicates the boundary constraint of neural wave diffusion when the AUV avoids dynamic obstacles. The set of points contained inside the closed elliptic curve is an optional neuron node. The density of the grid inside the black curve is significantly different from that of the green curve because of the intrusion of dynamic obstacles. The AUV completes

TABLE 2. Control parameters.

AUV-No	1	1	1	1	4	4	4	4
Step-length								
Density								
Iterate-No	18	19	20	21	13	14	15	16
τ/λ_{hw}	2.17/10.86	2.17/10.36	2.17/9.86	2.52/8.20	6.41/7.13	6.41/6.63	6.41/6.12	6.41/5.63
κ/λ_{lw}	7.83/4.73	7.83/4.59	7.83/4.46	7.48/4.43	3.59/8.84	3.59/8.47	3.59/8.09	3.59/7.68

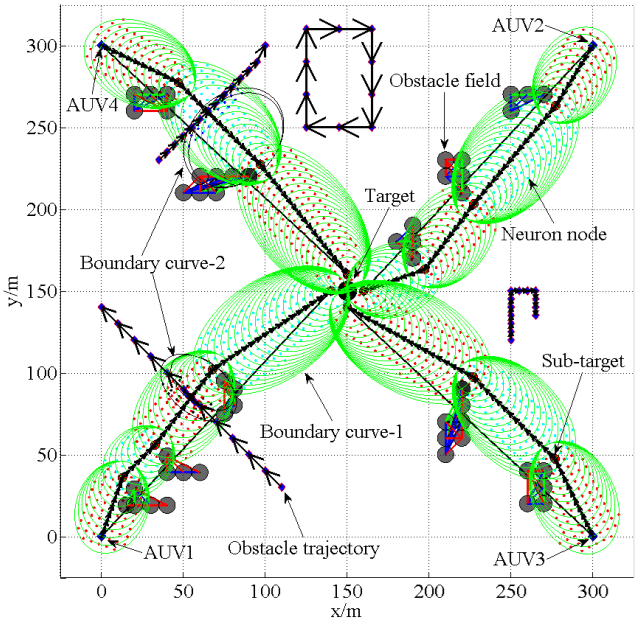

FIGURE 12. Operation chart without the interference of ocean currents.

the avoidance of dynamic obstacles under the constraint of the black curve. It is proven that dynamic obstacles can be avoided by changing the mesh density.

Real-time data are extracted during the operation of the multiple AUV system in the simulation experiment, as shown in the following Figure and table. Figure. 13 (a) and 13 (c) represent the horizontal and vertical density data curves of the grid during the collaborative operation of the AUV group respectively. Figure. 13 (b) and 13 (d) represent the horizontal and vertical step data curves of the AUV monomer, respectively. It can be concluded from Fig. 13 that the grid density is linearly distributed as the relative distance between the AUV and the capture point gradually shortens with the same step size. When a dynamic obstacle appears within the detection S-angle of the AUV at a certain speed, the mesh density and the step size of the AUV will change significantly and present a nonlinear distribution. The nonlinear distribution of the data curve provides the possibility for the AUV to achieve autonomous obstacle avoidance. The data of the nonlinear curve part are extracted as shown in Table 2.

After the environmental information is processed by the BNWN algorithm, the static obstacle group is divided into multiple domain spaces. This provides a basis for achieving the spatial recursion of subgoal nodes. The subtarget node information extracted from the simulation experiment is shown in Table 3:

TABLE 3. Control parameters.

No	1	2	3	4
$X_i(N)$				
Sub-target				
T_1	(12.69,35.82)	(277.31,263.18)	(276.82,47.31)	(46.82,277.31)
T_2	(32.69,55.82)	(227.31,203.18)	(226.82,97.31)	(96.82,227.31)
T_3	(67.69,101.82)	(197.31,163.18)	(150,140)	(150,160)
T_4	(140,150)	(160,150)		

From the simulation results, we can conclude that the multi-AUV system efficiently completes the tracking and capturing of the target based on the BNWN algorithm in a complex environment, and reduces the total length of the path and the number of turns, which fully reduces the energy consumption.

B. STATIC OCEAN CURRENT INTERFERENCE

This simulation experiment adds static ocean current interference based on the above research, which has a global impact. In this paper, the amplitude of the ocean current velocity is 2 and the angle with the X axis is 30 degrees (counterclockwise is positive). The simulation results are shown in Fig. 14. In actual working conditions, an AUV can measure the speed of the ocean current in real time according to its sensors. For this reason, the hybrid algorithm designed by this research has reserved an interface for the import of ocean current data.

Figure. 14 (a) shows that under the influence of ocean currents, the trajectory of the AUV has changed significantly, and the number of redirections and the total path length haves increased significantly. In addition, AUV1 collided with obstacles, which caused AUV1 to fail to proceed and to be declared scrap, which ultimately led to the failure of the mission. It can be seen that ocean current interference poses a great threat to the cooperative operation of multiple AUV systems. Therefore, to solve the above problems, this

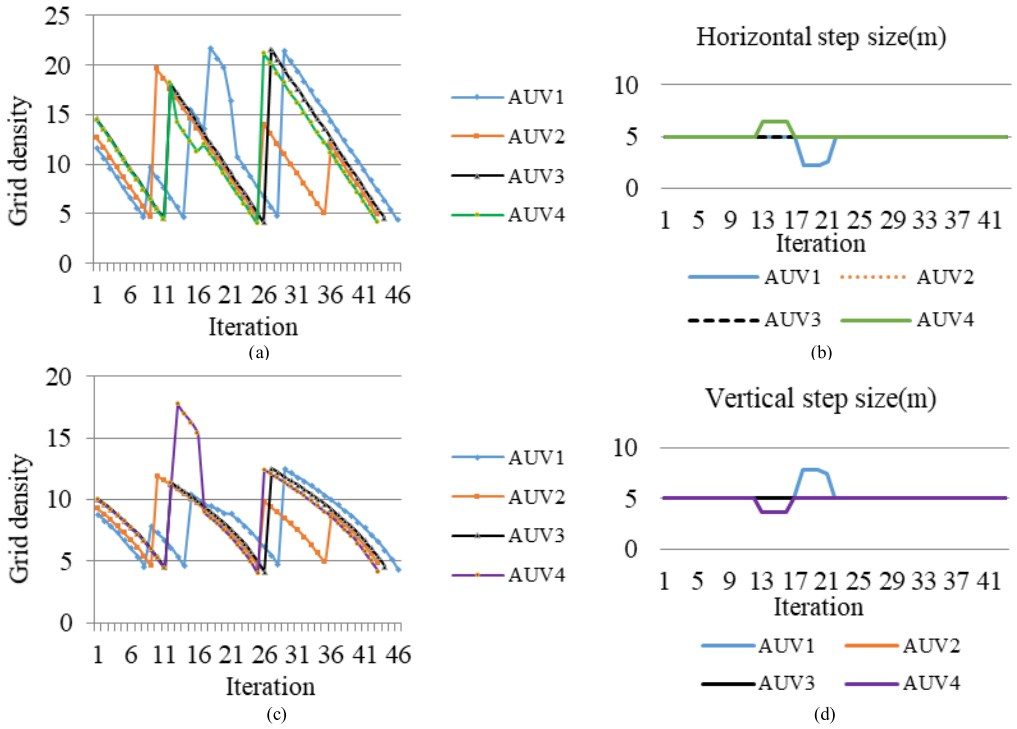

FIGURE 13. Data analysis.

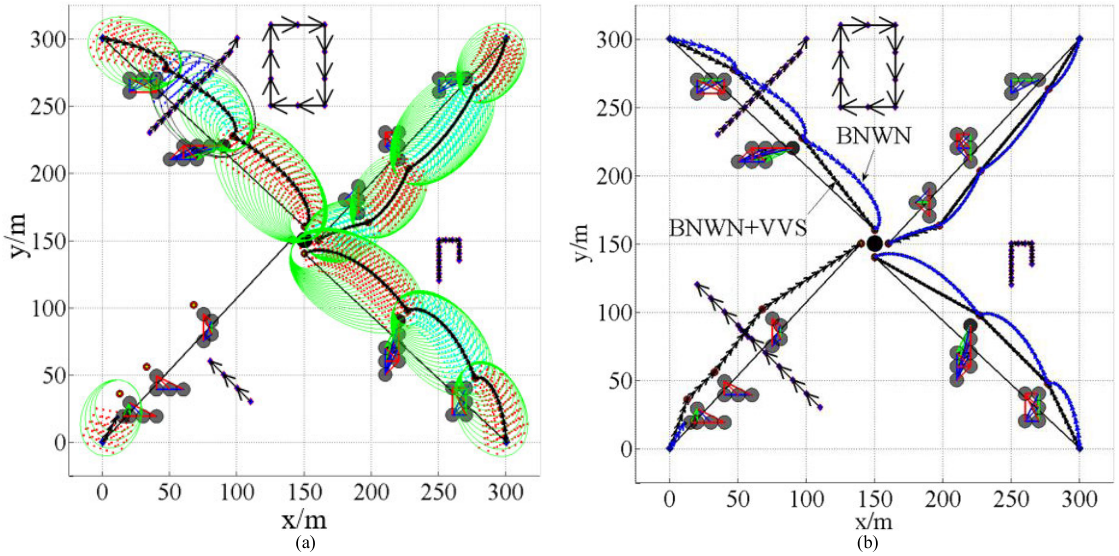

FIGURE 14. Operation process under the disturbance of steady ocean current.

paper proposes the VVS algorithm, which has low complexity and can achieve a significant reduction in response time. Comparison simulation results are shown in Fig. 14 (b). These results prove that the combination of the BNWN algorithm and the VVS algorithm can play a good role in optimizing the path, reducing the number of turns, and ensuring the success rate of tasks. Furthermore, the improvement of work efficiency and a reduction in total energy consumption are realized.

C. DYNAMIC OCEAN CURRENT INTERFERENCE

This experiment simulates complex sea conditions and mainly deals with target tracking and search under advanced sea conditions. Based on the above simulation experiments, time-varying ocean current interference is added. The time-varying curve of ocean current interference is shown in Fig. 15 (a), and the vector angle of ocean current velocity is expressed in radians. For time-varying ocean currents, this article assumes that the changes in the amplitude and angle of

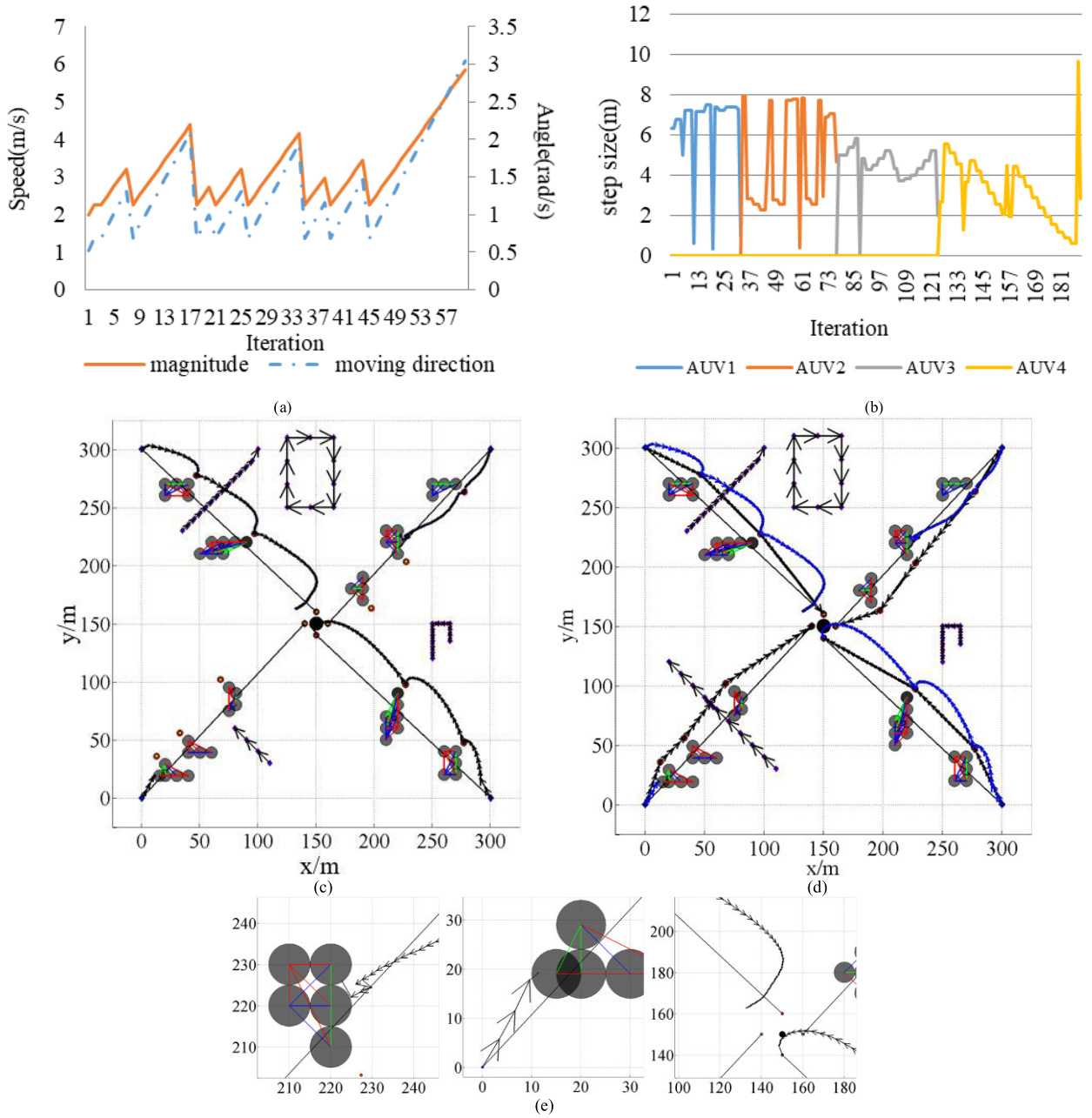

FIGURE 15. Real-time simulation results.

ocean currents follow non-fixed periodic linear incremental changes. Set the sub-target point as the inflection point of the linear interval, that is, the boundary of the non-fixed period. The update frequency of ocean current interference is set to 1/3 times the update frequency of AUV waypoints, that is, three waypoint updates correspond to one ocean current interference update. The simulation experiment results are shown in Fig. 15.

From Fig. 15 (c), it can be concluded that the travel trajectory of the AUV under the time-varying ocean current disturbance has changed more obviously than that under the static ocean current disturbance. As shown in Fig. 15 (e), under the interference of time-varying ocean currents, the collision rate of the AUV, and the loss rate of mission connection nodes will increase. Because of the simple structure, low complexity, short calculation time, and real-time update of the VVS algorithm and BNWN algorithm mentioned in this paper, this approach provides the basis for real-time control of the AUV trajectory path. It is concluded from Fig. 15 (d) that the VVS algorithm has the characteristic of keeping the trajectory path of the AUV consistent with the optimal trajectory path. The curves of the step size of the multi-AUV system are shown in Fig. 15 (b). The mutation in this Figure represents that the AUV successfully reaches the subtarget node.

The AUV updates and resets the graph node while it occupies the subtarget node. A comparison of Fig. 15 (c) and Fig. 15 (d) can verify the timeliness and accuracy of the algorithms proposed in this paper.

VII. COMPARATIVE ANALYSIS

For underwater robots, the complexity of the control algorithm, the length of the total path, and the smoothness of the trajectory are important indicators for evaluating the efficiency of the operation and the task completion rate.

In order to verify that the algorithm proposed in this paper has high efficiency and strong self-adaptive ability, multiple sets of comparative experiments will be conducted to verify the following. The selected comparison objects are the APF algorithm and A-star algorithm, both algorithms have representative significance.

The comparison simulation results are shown in Fig. 16.

As shown in Fig 16, the BNWN algorithm has the advantages of short path length, high smoothness, low collision rate, and strong stability compared to other algorithms. And the data extracted from the simulation experiment is shown in Table 4, which records the total length of the path and turn times base on different algorithms.

TABLE 4. Compare experimental data.

Algorithm Compare items	BNWN algorithm	APF algorithm	A-Star[26] algorithm
Path length	207.51	232.05	299.20
Turn times	3	14	19

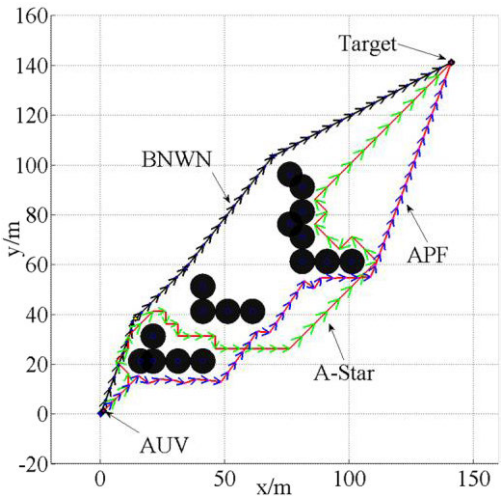

FIGURE 16. Simulation experiment comparison.

According to Table 4, the smoothness of the trajectory curve planned based on the BNWN algorithm is significantly higher than that of other algorithms, and the total length of the path is significantly shortened.

In addition, due to the establishment of the boundary function and the dynamic mesh model, the complexity of the BNWM algorithm is simplified and the stability of the BNWN algorithm is improved. Figure. 17 and Figure.18 show the working process of the A star algorithm and the BNWN algorithm under the same operating conditions.

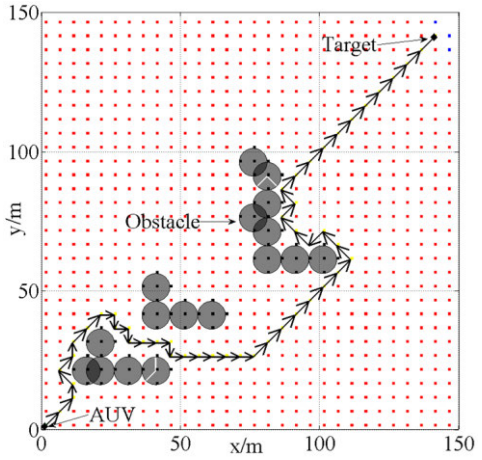

FIGURE 17. Path planning based on improved A-star algorithm.

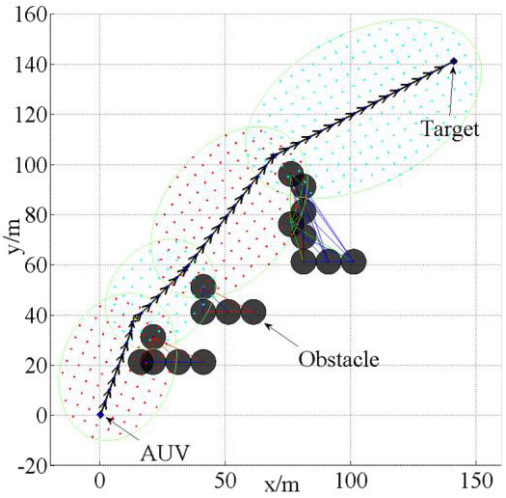

FIGURE 18. Path planning based on BNWN algorithm.

Obviously, under the condition of boundary function constraints, the neuron nodes required for real-time update and calculation in the BNWN algorithm are greatly reduced, and through the construction of dynamic grids, turn times are greatly reduced. The more complex the environment, the more obvious the advantages.

In addition, the BNWN algorithm can also enhance the stability of the algorithm by adjusting the control parameters of the obstacle domain. For example, to expand the influence range of the obstacle domain so that the sub-target nodes are far away from the obstacle space, which provides fault tolerance space for the errors caused by ocean current interference and sensor imbalance during the operation of the AUV individuals

In addition, we consulted a large number of documents related to this research and discovered a widely used

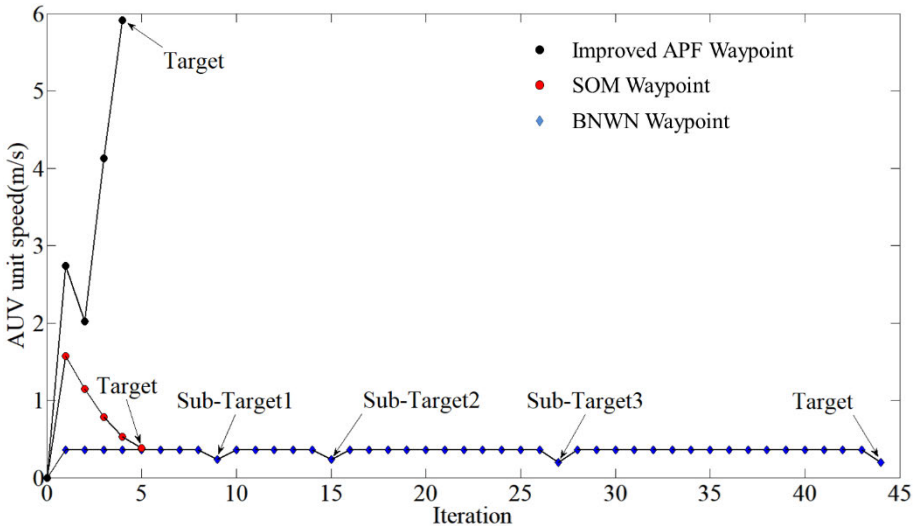

FIGURE 19. Comparative simulation test results between BNWN and SOM/Improved APF.

SOM algorithm [13], which has gradually become a research hotspot. However, the SOM algorithm and the improved A-star algorithm [26] have similar drawbacks, that is, there is a risk of speed jumping during the operation. In order to verify the accuracy of the algorithm proposed in this paper, we conducted a comparison simulation experiment between BNWN and SOM/Improved APF. In order to make the contrast effect more obvious, this article scales the speed accordingly. The result is shown in Fig. 19(Under the same simulation environment.):

In Fig. 19, the mutation point of the BNWN waypoint curve represents that the AUV has reached the sub-target point. In addition, it can be concluded from Fig.19 that the multi-AUV system based on the BNWN algorithm will not have the phenomenon of speed jump. Furthermore, the operation efficiency is improved and the energy consumption is reduced.

VIII. CONCLUSION

1) Decomposition of the workspace and space recursion of subtarget nodes realize the modularization of tasks.
2) The space sub-decomposition and the substitution of the constraint function construct a dynamic grid space, which has strong adaptability.
3) The DJ algorithm realizes autonomous judgment of a dynamic obstacle hazard field, prompting AUV individuals to avoid dynamic obstacle interference.
4) In a certain range, the combination of the BNWN algorithm and the VVS algorithm successfully overcomes the time-varying ocean current interference, which significantly reduces the number of redirections and the total path length.

REFERENCES

[1] Q. W. Liang, J. L. Ou, Z. X. Xue, and C. Ippolito, "Influences of temperature and salinity on holistic network performability of multi-AUV cooperative systems," *ISA Trans.*, vol. 93, pp. 165–171, Mar. 2019, doi: 10.1016/j.isatra.2019.03.014.
[2] S.-C. Yu, "Modeling of high-resolution 3d sonar for image recognition," *Int. J. Offshore Polar Eng.*, vol. 22, no. 3, pp. 186–192, Sep. 2012.
[3] J. Pyo, H. Cho, and S. C. Yu, "Beam slice-based recognition method for acoustic landmark with multi-beam forward looking sonar," *IEEE Sensors J.*, vol. 17, no. 21, pp. 7074–7085, Nov. 2017, doi: 10.1109/JSEN.2017.2755547.
[4] L. Yue and H. Chen, "Unmanned vehicle path planning using a novel ant colony algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 4, May 2019, doi: 10.1186/s13638-019-1474-5.
[5] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, Oct. 2017.
[6] Z. Q. Jiao, K. Ma, and Y. L. Rong, "A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs," *J. Comput. Sci.*, vol. 25, pp. 50–57, Mar. 2018, doi: 10.1016/j.jocs.2018.02.004.
[7] H. Liu, B. Xu, D. Lu, and G. Zhang, "A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm(Article)," *Appl. Soft Comput.*, vol. 68, pp. 360–376, Jan. 2015, doi: 10.1016/j.asoc.2018.04.015.
[8] J.-H. Liang and C.-H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm," *Adv. Eng. Softw.*, vol. 79, pp. 47–56, Jan. 2015, doi: 10.1016/j.advengsoft.2014.09.006.
[9] S. Hassan and J. Yoon, "Haptic assisted aircraft optimal assembly path planning scheme based on swarming and artificial potential field approach," *Adv. Eng. Softw.*, vol. 69, pp. 18–25, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.003.
[10] K. Ninomiya, M. Kapadia, A. Shoulson, F. Garcia, and N. Badler, "Planning approaches to constraint-aware navigation in dynamic environments," *Comput. Animation Virtual Worlds*, vol. 26, no. 2, pp. 119–139, Mar. 2015, doi: 10.1002/cav.1622.
[11] Q. Y. Pan and X. Y. Wang, "Independent travel recommendation algorithm based on analytical hierarchy process and simulated annealing for professional tourist," *Appl. Intell.*, vol. 48, no. 6, pp. 1565–1581, Jun. 2018, doi: 10.1007/s10489-017-1014-0.
[12] H. Duan and L. Huang, "Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning," *Neurocomputing*, vol. 125, pp. 166–171, Feb. 2014, doi: 10.1016/j.neucom.2012.09.039.
[13] D. Q. Zhu, Y. Liu, and B. Sun, "Task assignment and path planning of a multi-AUV system based on a Glasius bio-inspired self-organising map algorithm (Article)," *J. Navigat.*, vol. 71, no. 2, pp. 482–496, Mar. 2018, doi: 10.1017/S0373463317000728.
[14] J. Ni, X. Li, M. Hua, and S. X. Yang, "Bioinspired neural network-based Q-learning approach for robot path planning in unknown environments," *Int. J. Robot. Autom.*, vol. 31, no. 6, pp. 464–474, 2016, doi: 10.2316/Journal.206.2016.6.206-4526.
[15] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nucl. Eng. Des.*, vol. 326, pp. 79–86, Jan. 2018, doi: 10.1016/j.nucengdes.2017.11.006.

[16] B. Wang, S. Li, J. Guo, and Q. Chen, "Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm," *Neurocomputing*, vol. 282, pp. 42–51, Mar. 2018, doi: 10.1016/j.neucom.2017.12.015.
[17] X. D. Wu, W. B. Bai, and Y. E. Xie, "Planning approaches to constraint-aware navigation in dynamic environments," *Appl. SOFT Comput.*, vol. 73, pp. 735–747, Dec. 2018, doi: 10.1016/j.asoc.2018.09.011.
[18] A. M. Kashtiban and S. Khanmohammadi, "A genetic algorithm with SOM neural network clustering for multimodal function optimization," *J. Intell. Fuzzy Syst.*, vol. 35, no. 4, pp. 4543–4556, Oct. 2018, doi: 10.3233/JIFS-131344.
[19] S. J. Rasmussen and T. Shima, "Tree search algorithm for assigning cooperating UAVs to multiple tasks," *Int. J. Robust Nonlinear Control, IFAC-Affiliated J.*, vol. 18, no. 2, pp. 135–153, 2008.
[20] Z. R. Huang, D. Q. Zhu, and B. Sun, "A multi-AUV cooperative hunting method in 3-D underwater environment with obstacle," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 192–200, Oct. 2016.
[21] K. P. Cheng, R. E. Mohan, N. H. K. Nhan, and A. V. Le, "Graph theory-based approach to accomplish complete coverage path planning tasks for reconfigurable robots," *IEEE Access*, vol. 7, pp. 94642–94657, 2019, doi: 10.1109/ACCESS.2019.2928467.
[22] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1279–1298, Jul. 2008.
[23] X. Cao and D. Zhu, "Multi-AUV task assignment and path planning with ocean current based on biological inspired self-organizing map and velocity synthesis algorithm," *Intell. Autom. Soft Comput.*, vol. 23, no. 1, pp. 31–39, Jan. 2017, doi: 10.1080/10798587.2015.1118277.
[24] Q. Guo and D. Z. Zhang, "The path planning for robot soccer based on graph theory," *Harbin Gongye Daxue Xuebao*, vol. 42, pp. 18–20, Oct. 2010.
[25] X. Cao, H. Sun, and L. Guo, "Potential field hierarchical reinforcement learning approach for target search by multi-AUV in 3-D underwater environments," *Int. J. Control*, vol. 93, no. 7, pp. 1677–1683, Jul. 2020.
[26] B.-L. Zhang, "Development of path planning approach using improved a-star algorithm in AGV system(Article)," *J. Internet Technol.*, vol. 20, no. 3, pp. 915–924, 2017, doi: 10.3966/160792642019052003023.
[27] X. Fan, "Improved artificial potential field method applied for AUV path planning," *Math. Problems Eng.*, vol. 2020, pp. 1–21, Oct. 2020.

XIWEN MA received the B.E. degree in mechanical engineering from Nanjing Agricultural University, Nanjing, China, in 2018. He is currently pursuing the master's degree with Jilin University, Changchun, China.

His current research interests include cooperatively pursue and intelligent control of multiple autonomous underwater vehicles, fault-tolerant control, and modern fault diagnosis, optimizing, and controller design.

CHEN YANLI received the B.E. degree in mechatronic engineering from Jiamusi University, Jiamusi, China, in 2007, and the M.E. and Ph.D. degrees in mechatronics engineering from Jilin University, Changchun, China, in 2009 and 2012, respectively.

He was a Postdoctoral Research Worker in solid geophysics with Jilin University, in 2016, where he is a currently an Associate Professor. His research interests include cooperatively pursue and intelligent control of multiple autonomous underwater vehicles, fluid power transmission, and control.

GUIQIANG BAI received the B.E. degree in mechanical design manufacture and automation from Yantai University, Yantai, China, in 2016, and the M.E. degree in mechatronic engineering from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree with Jilin University, Changchun, China.

His research interests include recovery and intelligent control of UUV, cooperatively pursue, and intelligent control of multiple autonomous underwater vehicles.

JUN LIU (Student Member, IEEE) received the B.Eng. degree in computer science from Wuhan University, China, in 2002. He is currently pursuing the Ph.D. degree with the Underwater Sensor Network (UWSN) Laboratory, University of Connecticut. He is also working as a Research Assistant with the Underwater Sensor Network (UWSN) Laboratory, University of Connecticut. His major research interests include time synchronization, localization, deployment for underwater acoustic networks, operating systems, and cross-layer design. He is a Student Member of the IEEE Computer Society.