THE JOURNAL OF NAVIGATION (2018), **71**, 482–496. © The Royal Institute of Navigation 2017
doi:10.1017/S0373463317000728

Task Assignment and Path Planning of a Multi-AUV System Based on a Glasius Bio-Inspired Self-Organising Map Algorithm

Daqi Zhu, Yu Liu and Bing Sun

(*Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China*)
(E-mail: zdq367@aliyun.com)

For multi-Autonomous Underwater Vehicle (multi-AUV) system task assignment and path planning, a novel Glasius Bio-inspired Self-Organising Map (GBSOM) neural networks algorithm is proposed to solve relevant problems in a Three-Dimensional (3D) grid map. Firstly, a 3D Glasius Bio-inspired Neural Network (GBNN) model is established to represent the 3D underwater working environment. Using this model, the strength of neural activity is calculated at each node within the GBNN. Secondly, a Self-Organising Map (SOM) neural network is used to assign the targets to a set of AUVs and determine the order of the AUVs to access the target point. Finally, according to the magnitude of the neuron activity in the GBNN, the next AUV target point can be autonomously planned when the task assignment is completed. By repeating the above three steps, access to all target points is completed. Simulation and comparison studies are presented to demonstrate that the proposed algorithm can overcome the speed jump problem of SOM algorithms and path planning in the 3D underwater environments with static or dynamic obstacles.

KEYWORDS

1. Multi-AUV System. 2. GBNN. 3. SOM. 4. Task assignment. 5. Path planning.

Submitted: 22 August 2016. Accepted: 31 August 2017. First published online: 13 October 2017.

1. INTRODUCTION. Due to the limited energy supply and narrow range of observations of a single Autonomous Underwater Vehicle (AUV), many practical underwater exploration missions have gone beyond the ability of a single AUV. The multi-AUV system is a new research area, concentrating on the aspects of applicability, robustness and scalability. For the multi-AUV system (Luo et al., 2014; Javier et al., 2015; Cao and Zhu, 2015; Sahar et al., 2014; Jaillet and Porta, 2013), it is important to improve the efficiency of the underwater work with reasonable task assignment and path planning under the prevailing

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

conditions. Task assignment and path planning is a technique to control a set of AUVs to access a target point with an optimal path based on algorithms (Huang et al., 2016). There has been some research on task assignment and path planning of multi-AUV systems. Kulkarni and Pompili (2013) developed a task allocation framework for networked AUVs that work as a team to accomplish critical missions. Ren and Mei (2013) applied an ant colony algorithm to solve the task allocation problem so that an underwater glider deals with multiple tasks according to the tasks' situations. The above algorithms can be used to assign different tasks, but they do not have the ability to self-regulate. The efficiency is low because of limited collaboration between AUVs.

In recent years, self-organising neural networks have been widely used in the task assignment and path planning of multi-AUV systems because of their competitive characteristics and self-learning abilities. A self-organising neural network was first proposed by Kohonen (1998) and it was mainly used for dynamic regional classification of the input vector. When it is applied to task assignment of multiple robots, the targets will be included, divided and assigned automatically to the most suitable robot. Zhu and Yang (2006; 2012) applied a self-organising neural network to the task assignment and path planning of a multi-robot system. The algorithm they proposed can make each robot reach the target along the shortest path under the premise that the multi-robot system covers all the targets with minimal total energy consumption and can effectively solve the situation that the number of robots and the number of targets are unknown. Aluizio and Santana (2015) proposed a Self-Organising Map-State Trajectory Generator (SOM-STG) to plan and control legged robot locomotion. The SOM-STG can construct one or more gaits for a simulated robot with six legs and control the robot with any of the gaits learned, and can smoothly swap gaits. In addition, the SOM-STG can learn to construct a state trajectory from observing an animal in locomotion. All of the methods above are for ground mobile robots, and the research of task assignment and path planning for a multi-AUV system is rare.

Inspired by the applications of the SOM algorithm in multi-agent systems, Huang et al. (2012), Zhu et al. (2013) and Huang et al. (2014) proposed an integrated multi-AUV dynamic task assignment and path planning algorithm by combining the Self-Organising Map (SOM) neural network and a novel velocity synthesis approach. Task assignment and path planning of a multi-AUV system for static and dynamic objects is discussed within a Three-Dimensional (3D) time-varying ocean environment. This algorithm achieves path planning and navigation through the optimisation and equalisation of each AUV's workload and weight updates by SOM. Although the algorithm in the literature (Huang et al., 2012; Zhu et al., 2013; Huang et al., 2014) can solve task assignment and path planning of a multi-AUV system, it does not consider obstacles in real environments. Additionally, each AUV speed is based on the weight function of the SOM and is unlimited, allowing the AUVs' control constraints to be exceeded. This is often referred to as the speed jump problem. When the target is far from the AUV, the SOM solution proposes that the AUV travels a few kilometres or even several tens of kilometres in each navigation step, which also exceeds the AUV's speed or range capability.

Aiming at the problem of speed jump and the absence of an obstacle avoidance capability in the SOM algorithm, a Glasius Bio-inspired Self-Organising Map (GBSOM) (Glasius et al., 1995) algorithm combined with a Glasius Bio-inspired Neural Network (GBNN) (Luo et al., 2014a; 2014b) and SOM is proposed in this paper. The implementation steps are: (1) 3D GBNN model is established to represent the grid of the 3D underwater working

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

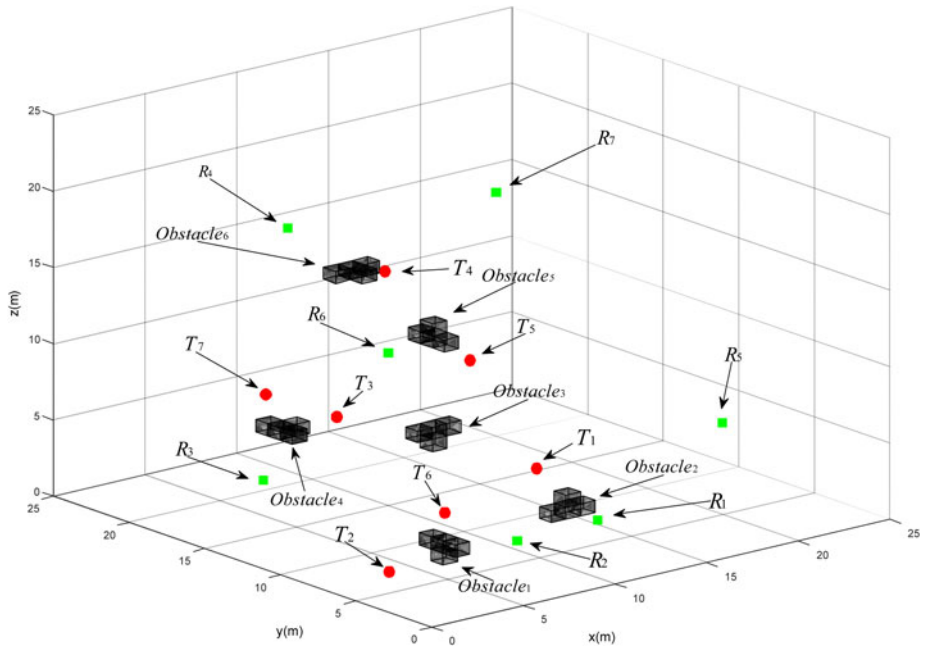


Figure 1. 3-D underwater workspace.

environment. The magnitude of the neural activity at each node of the GBNN is calculated. (2) SOM is used to assign the targets to a set of AUVs and determine the order of the AUVs accessing the target point. (3) The AUVs' next navigation point can be planned autonomously according to the distribution of the neuron activity in the GBNN. (4) GBNN is updated. By repeating steps (2), (3) and (4), access to all target points can be completed. The GBSOM algorithm is not only able to prioritise the task assignment, but can also effectively plan a collision-free shortest path for the AUV.

The organisation of the paper is as follows. Section 2 defines the multi-AUV system problem to be solved. The traditional SOM algorithm is introduced in Section 3 and an improved GBSOM algorithm based on SOM is then introduced in Section 4. Section 5 provides simulation results and evaluation of the proposed GBSOM algorithm and Section 6 gives the concluding remarks.

2. THE MULTI-AUV SYSTEM PROBLEM.

The core issue of a multi-AUV system is to minimise its energy cost. There are two main aspects of the energy loss in the multi-AUV system: task allocation, and AUV path planning. Figure 1 is a schematic diagram of a 3D underwater environment with obstacles, AUVs and targets. In this paper, we assume that the locations of obstacles, targets and AUV are known.

To minimise the energy cost of the multi-AUV system, a reasonable task allocation is first given. For each AUV the need to reach the target and the order of arrival is determined and then the optimal path is planned. The path of each AUV must be collision-free, and the distance and speed of each AUV must meet the kinematics requirements.

The subject of this paper is the task allocation and path planning of a multi-AUV system. The task allocation problem includes one-to-one and one-to-many. The path planning

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

problem includes obstacle avoidance and speed hopping. In this paper, the initial positions of AUVs and targets are required to meet the following principles: The initial location of the AUVs and targets needs to contain all the situations mentioned in this article (one-to-one and one-to-many task assignment and obstacle avoidance). Therefore, in the design of the simulation environment, the initial position of the AUV and the target is not randomly assigned: the locations of AUVs and the targets are chosen to model different task assignment and obstacle avoidance problems. For example, for the task assignment in Figure 1, R2 and R3 might have two corresponding targets and R1, R4 and R6 might be considered to have only one corresponding target. All task assignments are presented in the simulation environment. For path planning, there is no linear path between each target and its corresponding AUV in the designed simulation environment to meet the collision-free requirement. In this way the design of the initial environment can test the performance of the proposed GBSOM algorithm in this paper.

3. TRADITIONAL SOM TASK ASSIGNMENT ALGORITHM. A SOM algorithm is an unsupervised clustering algorithm, which does not need to be trained in advance. It can automatically cluster data according to a certain type of attribute. The distance between an AUV and the target can be used as a clustering attribute in multi-AUV task assignment. The traditional SOM algorithm is composed of two layers of neurons, which are the input and output layers. In the input layer, the external information is collected at each neuron of the output layer through a weight vector. The competitive layer (i.e. the output layer) is responsible for the comparison and analysis of input patterns, finding the rules and classifying them.

In multi-AUV task assignment, the coordinates of the target are selected as the input layer, and the coordinates of the current AUV is selected as the output layer. The network structure of SOM is shown in Figure 2. The SOM algorithm can be divided into the following parts: (1) the selection of the winning neuron; (2) the determination of the neighbourhood function; (3) the update of the weight.

3.1. *The selection of the winning neuron.* The principle of the winning neuron selection is the most similar to the clustering attribute. The Euclidean distance of the attribute is usually used to represent the degree of approximation of the attribute; the smaller the Euclidean distance the more similar the attribute. The winner selection rule is obtained by the following expression:

$$[P_j] \Leftarrow \min(D_{jl}, j = 1, 2, 3, \ldots, J; l = 1, 2, 3, \ldots, L) \tag{1}$$

where $[P_j]$ denotes that the j-th neuron is the selected winner to the l-th input neuron. D_{jl} represents the Euclidean distance between the j-th output node and the l-th input neuron. The Euclidean distance calculation between two neurons is defined as:

$$|T_l - R_j| = \sqrt{(x_l - w_{jx})^2 + (y_l - w_{jy})^2 + (z_l - w_{jz})^2} \tag{2}$$

$T_l = (x_l, y_l, z_l)$ represents the coordinates of the l-th input neural node which represents the target location. $R_j = (w_{jx}, w_{jy}, w_{jz})$ are the coordinates of output neurons which represent one specific AUV location.

3.2. *The determination of the neighbourhood function.* When the winning neuron of the SOM network is selected, the next step is to determine the neighbourhood function.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

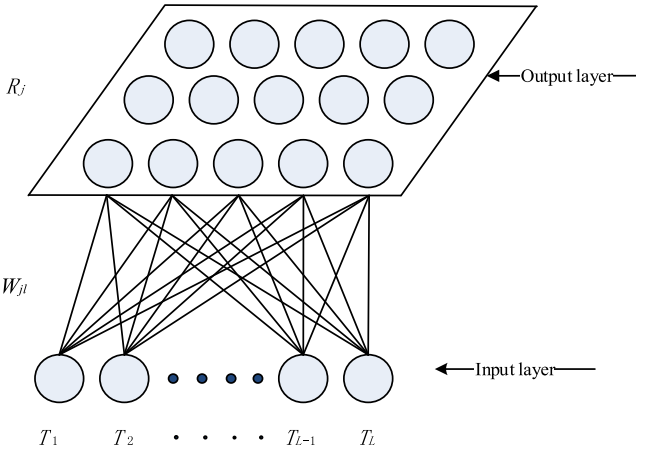


Figure 2. The network structures of SOM.

The neighbourhood function determines the degree of attraction of the input neurons to the winning neuron and its neighbourhood. The Gauss function is generally selected as the neighbourhood function, which is described as follows:

$$f(d_m, G) = \begin{cases} e^{-d_m^2/G^2(t)}, & d_m \le \lambda \\ 0, & d_m > \lambda \end{cases} \tag{3}$$

$$G(t) = (1 - \beta)^t G_0 \tag{4}$$

where $d_m = |N_m - N_j|$ represents the distance between the *m-th* output neuron and the winning neuron j. λ is the neighbourhood radius and β is the decay rate. The convergence time of the SOM neural network is determined by the value of β and is proportional to its magnitude. The AUV navigation step size in the neighbourhood can be controlled by setting the value of β and G_0, which can control the accuracy of AUV navigation.

The neighbourhood function is determined according to the actual situation. When one task requires only one AUV, λ should be selected to be 0 to use the AUV efficiently. When one task requires multiple AUVs at the same time, the value of λ should be selected as a suitable constant.

3.3. *The update of the weight.* In this paper, the weight of the SOM equals the position of the AUV. When the neighbourhood function is determined, the last step of the SOM algorithm is to update the weights. The purpose of the weight updating is to make the SOM algorithm converge, which is to make the AUV close the target point in the multi-AUV system task assignment.

$$R_j(t+1) = \begin{cases} R_j(t) + \alpha \times f(d_m, G) \times (T_l - R_j(t)), & D_{jl} > D_{\min} \\ T_l, & D_{jl} \le D_{\min} \end{cases} \tag{5}$$

$R_j = (w_{jx}, w_{jy}, w_{jz})$ represents the position coordinates of the j-th AUV. $T_l = (x_l, y_l, z_l)$ represents the position coordinates of the l-th target. α is the learning rate and the range of α is $0 < \alpha \le 1$. $D_{\min}$ represents the minimum distance from the AUV to the target. When the

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

distance between the AUV and the target is less than or equal to $D_{\min}$, the AUV has arrived at the target. D_{jl} is a relevant Euclidean distance value between the winning AUV and the corresponding target position.

From the above description, the traditional SOM algorithm has been found to have the following shortcomings. First, the path planned by the SOM algorithm is a straight line, so the SOM algorithm cannot avoid obstacles effectively. Second, since the weight update function of the SOM corresponds to the navigational distance between the AUV and the target from using Equation (5), in practical applications there is a problem of speed jumps: the initial update may be so large as to generate a speed requirement beyond that of the AUV. In reality, during the early iterations of the SOM algorithm, the AUV might be asked to sail many kilometres in a very short period of time; far exceeding the AUV's speed and range capability.

4. THE GLASIUS BIO-INSPIRED SELF-ORGANISING MAP (GBSOM).

In order to avoid these two problems of traditional SOM algorithms, a 3D GBNN model is established to represent the grid of the 3D underwater working environment. The corresponding dynamic neural network is built on the grid map. Each point in the grid map corresponds to the neural activity of the neurons in the GBNN. The GBNN model is applied to regulate the weights of SOM. The process can be described as follows: select the maximum neural activity point in the AUV's neighbourhood as *Pn*, and then put *Pn* into Equation (12) to update the weight of SOM. An integrated GBSOM method is proposed in this section. This section can be divided into two parts. The GBNN algorithm is introduced in Section 4.1, and the combination of GBNN and SOM is introduced in Section 4.2.

4.1. *GBNN model.*

The GBNN algorithm in this paper is a kind of discrete Hopfield-type neural network (Hopfield and Tank, 1985). Its kinetic model is as follows:

$$x_i(t+1) = g\left(\sum_{j=1}^{J} \omega_{ij} \times x_j(t) + I_i\right) \tag{6}$$

$$\omega_{ij} = \begin{cases} e^{-\gamma |i-j|^2}, & |i-j| \le r \\ 0, & |i-j| > r \end{cases} \tag{7}$$

$$g(x) = \begin{cases} 0, & x < 0 \\ \beta x, & x \in [0,1] \\ 1, & x > 1 \end{cases} \tag{8}$$

In Equation (6), $x_i(t+1)$ represents the neural activity of the i-th neuron in the neural network at time instance $t+1$, $x_j(t)$ is the neural activity of the neuron j which side-connects with the neuron i with time instance t. In Equation (7), ω_{ij} is a symmetric connection coefficient matrix, which is a connection weight coefficient between neuron i and neuron j. $|i-j|$ is the Euclidean distance between neuron i and neuron j. r is a constant value that is more than zero, and the value of r is set as $\sqrt{3}$ for the 3D neural network in this paper. The 2D distribution of ω_{ij} is shown in Figure 3. In Figure 3, r is a regional radius. Each neuron in the neural network has a right angle or a diagonal connection to the neurons in the region of r. Figure 3 also illustrates the winning point selected from the eight neighbourhoods around the AUV which is seen as a temporary target for the 2D neural network. In the same way, the number of neighbourhoods is 26 for the 3D neural network.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

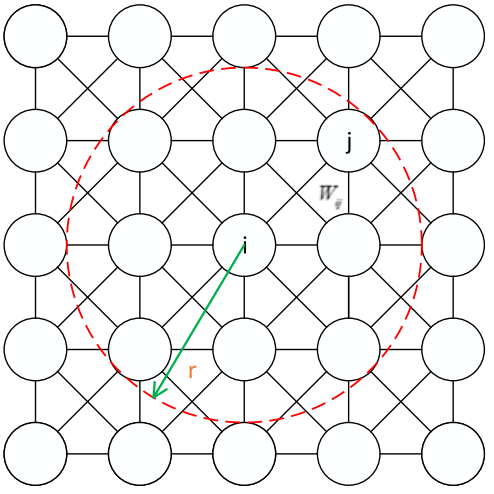


Figure 3. Two-dimensional neural network.

Equation (8) is the transfer function, which limits the output range of the kinetic model, ensuring that the neural activity of the neuron is guaranteed between 0 and 1 and makes the system in a stable state. I_i represents the additional input of the i-th neuron. When the neuron is the target, I_i is $E(E>=1)$. When the neuron is an obstacle, I_i is $-E$. Otherwise, I_i is 0.

$$I_i = \begin{cases} E, & \text{if it is a target} \\ -E, & \text{if it is an obstacle} \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

For the AUV's current position in the GBNN denoted as P_i, P_k is the coordinate of the points in k-th P_i's neighbourhood, and the next moving position P_n of AUV can be obtained by

$$P_n = \arg\max\{x_{P_k}, P_k \in \overset{\circ}{U}(P_i, r)\} \tag{10}$$

Since the neural activity of target positions are positive in the GBNN and the neural activity of obstacle positions are negative, the target positions attract the AUV globally while the obstacle areas just push the AUV away to avoid local collisions. For more detail, the reader can refer to Luo et al. (2014a; 2014b).

4.2. *The combination of GBNN and SOM.* On the one hand, the distance between the target point and AUV is seen as the competitive attribute in traditional SOM algorithms. The path planned by the SOM algorithm is a straight line, so the traditional SOM algorithm cannot avoid obstacles in practice. On the other hand, the update strategy of the SOM algorithm is successive approximation, and the step length of each update is gradually reduced. In practice, the step size updated is very large at the initial time, which causes the speed jump problem.

Task assignment and path planning are continuously implemented in the GBSOM algorithm, until the corresponding AUV reaches the corresponding target. The flow chart of the GBSOM algorithm is shown in Figure 4.

The specific steps of the GBSOM algorithm are as follows:

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

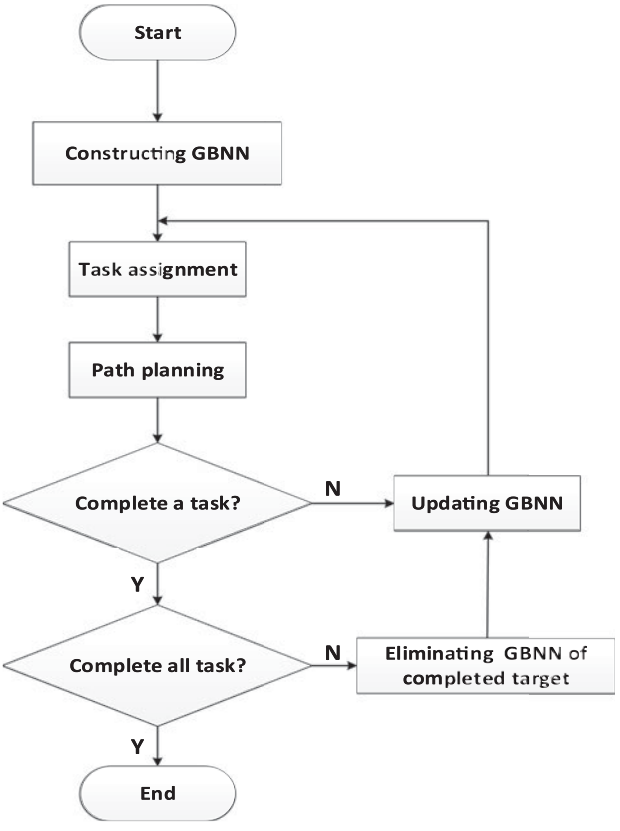


Figure 4. Flow chart of the GBSOM algorithm.

(1) A corresponding GBNN is constructed for each target point.
(2) Task assignment is implemented.

AUV Task assignment can be divided into two categories. The first is one AUV corresponding to only one target. The second is one AUV corresponding to multiple targets. The first case is relatively simple and only needs to match one AUV with one target. For the second case, due to one AUV corresponding to multiple targets, the order of the corresponding targets should be determined based on the shortest total distance after the magnitude of the neural activity for each AUV/target pair has been assessed and the relevant AUV/target combination chosen; that is, task assignment has been conducted. Equation (11) is used to complete the task assignment. $V(P_{T_l}, P_{R_j})$ represents the neural activity of the j-th AUV in the GBNN of l-th target, where R_j represents the final winning AUV corresponding to the target T_l.

$$(T_l, R_j) \Leftarrow \arg\max \max(V(P_{T_l}, P_{R_j})) \quad l = 1, 2, 3, \ldots, L; \quad j = 1, 2, 3, \ldots, J \tag{11}$$

(3) Path planning is performed on the basis of the completion of task assignment.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

After the AUV's next motion direction location P_n is obtained by Equations (6) and (10), to avoid the sharp speed jumps of AUV in SOM path planning, the weight of the SOM is updated:

$$R_j(t+1) = \begin{cases} R_j(t) + \alpha \times f(d_m, G) \times (P_n - R_j(t)), & D_{jl} > D_{\min} \\ P_n, & D_{jl} \le D_{\min} \end{cases} \tag{12}$$

$P_n = (x_n, y_n, z_n)$ are the coordinates of the AUV in the next moving direction and $R_j = (w_{jx}, w_{jy}, w_{jz})$ is the SOM output neuron which represents the AUV position. $R_j(t)$ represents the coordinates of the AUV at the present time, $R_j(t+1)$ represents the coordinates of the AUV at the next time marker. Compared with the update rule for the winning AUV in Equation (5), as the distance between neighbouring grids is short in the grid map, and AUVs always move toward a neighbouring grid of their current location based on Equation (12), $P_n - R_j(\mathrm{t})$ must be the appropriate value and meet the AUV's speed control constraint (unit speed one). Therefore, AUVs do not experience sharp speed jumps. In addition, due to the GBNN inhibiting the AUV from hitting obstacles, the path planning method based on GBNN can avoid either static or dynamic obstacles. Thus, the proposed GBSOM multi-AUV task assignment and path planning algorithm is valid in an underwater environment with random motion of target location and obstacles.

(4) Determine whether the task is completed.

If the task is completed, then the mission is over. Otherwise, the GBNN is updated and the second and third steps are repeated.

5. SIMULATION STUDIES. The simulation of the underwater environment takes place in a 25 metre 3D grid map with the length of each grid cube set to 1 metre. A corresponding neural network structure is built for each target on the grid map and the neural activities of all neurons are initialised to 0. In addition, the shape and size of the AUV are not considered in the simulation, which means it is simplified as a particle. All the chosen parameters of the GBSOM algorithm in the simulation experiments are listed in Table 1.

In order to test the effectiveness of the algorithm, the simulation results of the task assignment and path planning of a multi-AUV system are shown below.

5.1. *The task assignment and path planning in a 3D static environment.*

5.1.1. *One AUV corresponds to one target.* The amount of AUVs is equal to the number of targets, which means one AUV is assigned to one target. The simulation results of the task assignment and path planning of multi-AUV in a 3D static environment are shown in Figure 5. The initial positions of the AUVs are: $R_1(10, 5, 1), R_2(5, 5, 3), R_3(15, 15, 6), R_4(20, 20, 10)$. The initial positions of targets are: $T_1(1, 5, 1), T_2(13, 6, 7), T_3(10, 15, 3), T_4(12, 20, 15)$.

One AUV corresponds to one target, which is the simplest and most basic case. Only one SOM competition is needed to complete the AUV task assignment. The initial task assignment is completed by selecting the AUV with the highest neural activity in the GBNN of each target.

Table 2 shows the neural activity of each AUV in the GBNN of each target at the initial state in task assignment, and the red region is the maximum activity for each T_l. The initial task assignment is accomplished by selecting the AUV whose neural activity

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

Table 1. Control parameters.

Parameters	Value	Description
G_0	0·8	a positive constant
β	0·1	the update rate determining the calculation time
$D_{\min}$	0·3	the minimum distance
α	0·3	The learning rate of SOM
γ	2	a positive constant
r	$\sqrt{3}$	radius of neighborhood
E	100	a positive constant

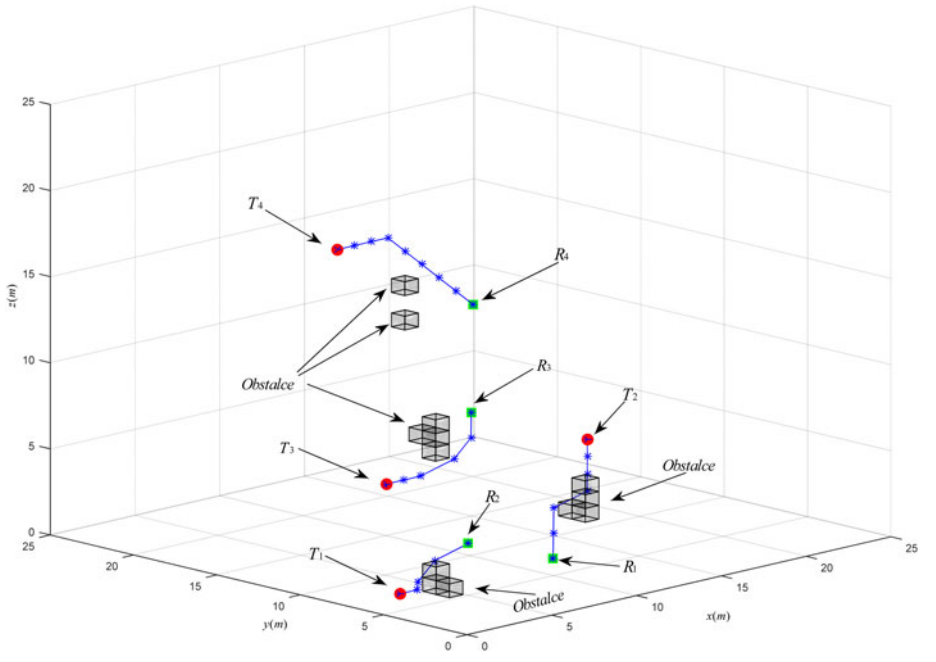


Figure 5. The simulation diagram of one AUV corresponding to one target in 3D.

is maximal in the GBNN of each target. The following briefly takes T_1 as an example to analyse the task assignment. From Table 2, the neural activities of four AUVs in the GBNN of T_1 are: $V(P_{R_1}, P_{T_1}) = 2{\cdot}04E\text{-}14$, $V(P_{R_2}, P_{T_1}) = 5{\cdot}95E\text{-}7$, $V(P_{R_3}, P_{T_1}) = 4{\cdot}03E\text{-}42$, $V(P_{R_4}, P_{T_1}) = 1{\cdot}03E\text{-}28$.

Since the neural activity of $V(P_{R_2}, P_{T_1})$ is maximal, T_1 is assigned to R_2. Similarly, the results of the initial assignment for the remaining target points are T_2-R_1, T_3-R_3, and T_4-R_4. After the completion of task assignment, the specific path of the AUV needs to be planned. The following briefly takes T_1-R_2 as an example to analyse the path planning.

Table 3 shows the GBNN neural activity in the 26 neighbourhoods of R_2, and the red region is the maximum activity in the 26 neighbourhoods of R_2. The P_n point whose neural activity is maximal among the 26 neighbourhoods of R_2's current position is selected as the next point in path planning. Firstly, P_n is (4, 6, 2) computed by Equation (10). Secondly, P_{n}

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

Table 2. The neural activity of each AUV in the GBNN of each target at initial state.

R_j / $V(P_{R_j}, P_{T_l})$	R_1	R_2	R_3	R_4
T_1	2·04E-14	5·95E-07	4·03E-42	1·03E-28
T_2	1·44E-10	4·08E-14	1·30E-25	1·72E-14
T_3	1·13E-33	2·87E-33	4·14E-15	6·85E-27
T_4	4·65E-16	7·51E-18	3·03E-21	2·97E-09

Table 3. The GBNN neural activity in the 26 neighbourhoods of R_2.

P_k	(4,4,2)	(4,4,3)	(4,4,4)	(4,5,2)	(4,5,3)	(4,5,4)	(4,6,2)
X_{p_k}	3·8528E-07	7·9212E-08	8·0977E-09	1·1300E-07	1·0111E-07	1·3350E-08	5·9538E-07
P_k	(4,6,3)	(4,6,4)	(5,4,2)	(5,4,3)	(5,4,4)	(5,5,2)	(5,5,4)
X_{p_k}	8·6221E-08	8·2884E-09	7·8279E-09	2·3359E-09	3·5141E-10	5·1355E-09	4·9540E-10
P_k	(5,6,2)	(5,6,3)	(5,6,4)	(6,4,2)	(6,4,3)	(6,4,4)	(6,5,2)
X_{p_k}	1·5623E-08	2·8827E-09	3·7447E-10	1·7325E-10	6·4024E-11	1·2566E-11	1·7803E-10
P_k	(6,5,3)	(6,5,4)	(6,6,2)	(6,6,3)	(6,6,4)		
X_{P_k}	7·4219E-11	1·6501E-11	4·0825E-10	8·8980E-11	1.4208E-11		

is taken into Equation (12), which means the specific path panning of P_n-R_2 is completed. Repeat these two steps until R_2 reaches the corresponding T_1.

5.1.2. *One AUV corresponds to multiple targets.* The simulation results of the task assignment and path planning of one AUV corresponding to multiple targets in a 3D static environment are shown in Figure 6. The initial positions of the AUVs are: $R_1(16, 4, 1), R_2(8, 4, 2), R_3(4, 16, 3), R_4(7, 18, 18), R_5(20, 5, 6), R_6(10, 15, 10), R_7(20, 20, 16)$. The initial positions of targets are: $T_1(14, 10, 3), T_2(1, 4, 2), T_3(8, 16, 6), T_4(9, 14, 16), T_5(12, 12, 10), T_6(4, 4, 5), T_7(5, 17, 8)$. For the second case in AUV task assignment, one competition can only determine to which targets each AUV corresponds, but cannot determine the order of AUVs in reaching the target points. The order of arrival should be determined in order to complete the task assignment, so it can be determined by adopting the second competition which is shown in Equation (11).

The following takes R_2 as an example to briefly analyse. First, the AUV whose neural activity is maximal in its GBNN is chosen for each target, the results of the selection are shown in the red region of Table 4. From Table 4, it can be seen that R_2 corresponds to T_2 and T_6, and the neural activity of R_2 in the *GBNN* of T_2 and T_6 are $V(R_2, T_2) = 5{\cdot}37E\text{-}11, V(R_2, T_6) = 2{\cdot}25E\text{-}7$. Inputting the above information into Equation (11), it is found that R_2 reaches T_6 first and then reaches T_2. Similarly, the results of the initial assignment for the remaining target points are R_1-T_1, R_2-T_6-T_2, R_3-T_7-T_3, R_4-T_4, R_6-T_5; R_5 and R_7 do not perform a task. After the completion of task assignment, the specific path of the AUV needs to be planned. The following briefly takes T_7-R_3 as an example to analyse the path planning. Table 5 shows the GBNN neural activity in the 26 neighbourhoods of R_3, and the red region is maximum activity in the 26 neighbourhoods of R_3. The point P_n whose neural activity is maximal among the 26 neighbourhoods of R_3 of the current position is selected as the next point in path planning. Firstly, P_n is (4, 17, 4) computed by Equation (10). Secondly, P_n is taken into Equation (12), which means the

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

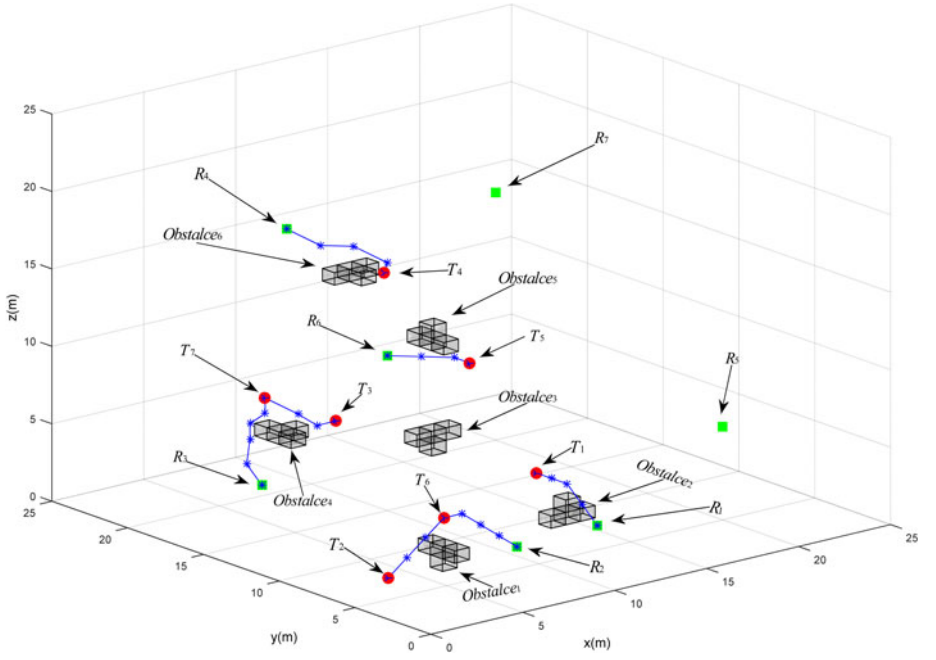


Figure 6. 3D simulation diagram of one AUV corresponding to multiple targets.

Table 4. The neural activity of each AUV in the GBNN of each target at the initial state.

R_j / $V(P_{R_j}, P_{T_l})$	R_1	R_2	R_3	R_4	R_5	R_6	R_7
T_1	6·44E-07	2·17E-13	1·07E-18	7·82E-30	7·12E-13	8·57E-15	2·90E-28
T_2	1·58E-20	5·37E-11	4·62E-19	1·20E-35	1·69E-31	3·01E-26	7·69E-47
T_3	2·65E-20	3·17E-20	2·05E-07	2·42E-19	1·26E-26	1·86E-06	4·86E-26
T_4	7·45E-29	3·96E-28	1·01E-22	2·15E-07	3·62E-28	2·72E-09	3·38E-20
T_5	1·93E-17	4·52E-19	5·31E-18	8·81E-18	5·65E-18	6·62E-05	2·06E-20
T_6	2·95E-17	2·25E-07	2·43E-19	1·72E-31	1·07E-25	1·54E-21	3·16E-41
T_7	1·00E-25	1·80E-23	2·42E-08	4·16E-16	1·73E-31	2·12E-08	7·64E-28

specific path panning of $P_n - R_3$ is completed. Repeat these two steps until R_3 reaches the corresponding T_7.

Comparison in terms of AUV speed for R_3 between SOM and GBSOM algorithm is shown in Table 6 and Figure 7 (blue lines show the GBSOM algorithm and the red lines show the SOM algorithm). The comparison shows that R_3 stops after only five iterations of the SOM algorithm alone; R_3 collides with *Obstacle*$_4$. In Table 6 and Figure 7, it is clear that the speeds of SOM navigation are far beyond the AUV's control constraint, but the proposed GBSOM algorithm fully meets these. The next motion direction of the AUV is towards the neighbouring coordinate of its current location based on the proposed GBSOM algorithm. The mobile speed of the AUV must meet its control constraint requirements

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

Table 5. The GBNN neural activity in the 26 neighbourhoods of R_3.

P_k	(3,15,2)	(3,15,3)	(3,15,4)	(3,16,2)	(3,16,3)	(3,16,4)	(3,17,2)
X_{pk}	8·2060E-13	2·5696E-11	7·3250E-10	2·6384E-12	9·3903E-11	3·1380E-09	4·5107E-12
P_k	(3,17,3)	(3,17,4)	(4,15,2)	(4,15,3)	(4,15,4)	(4,16,2)	(4,16,4)
X_{pk}	1·7127E-10	6·2146E-09	1·6294E-12	5·7859E-11	1·9829E-09	5·8987E-12	1·0236E-08
P_k	(4,17,2)	(4,17,3)	(4,17,4)	(5,15,2)	(5,15,3)	(5,15,4)	(5,16,2)
X_{pk}	1·0236E-08	5·1865E-10	2·4209E-08	9·9492E-13	2·8333E-11	6·9548E-10	3·7740E-12
P_k	(5,16,3)	(5,16,4)	(5,17,2)	(5,17,3)	(5,17,4)		
X_{pk}	1·2210E-10	3·4503E-09	9·2598E-12	3·4459E-10	1·1621E-08		

Table 6. Comparison of the first ten terms of speed for R_3 between SOM and GBSOM algorithms.

Algorithm \ Iterations	1	2	3	4	5	6	7	8	9	10
SOM	1·56	1·09	0·76	0·53	0·37					
GBSOM	0·34	0·37	0·35	0·36	0·31	0·34	0·37	0·35	0·37	0·38

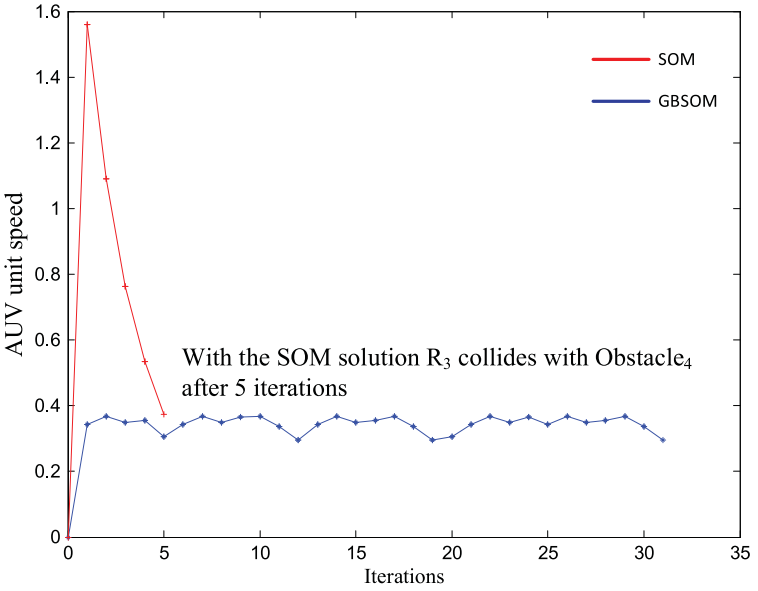


Figure 7. Comparison in terms of speed for R_3 between SOM and GBSOM algorithms.

and the AUVs have no sharp speed jumps. However, the traditional SOM task assignment and path planning algorithms do not consider the constraint of AUV speed, thus, AUV speed jumps will occur when the error T_l-R_j between the AUV's current coordinates R_j and corresponding target coordinates T_l suddenly change. In particular, the distance between the AUV and the target may reach tens of kilometres at the start times. Equation (5) is used to update the AUV position. The AUV is required to provide a very high speed to keep the algorithm real-time. It is practically impossible for an AUV to achieve such a high speed.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

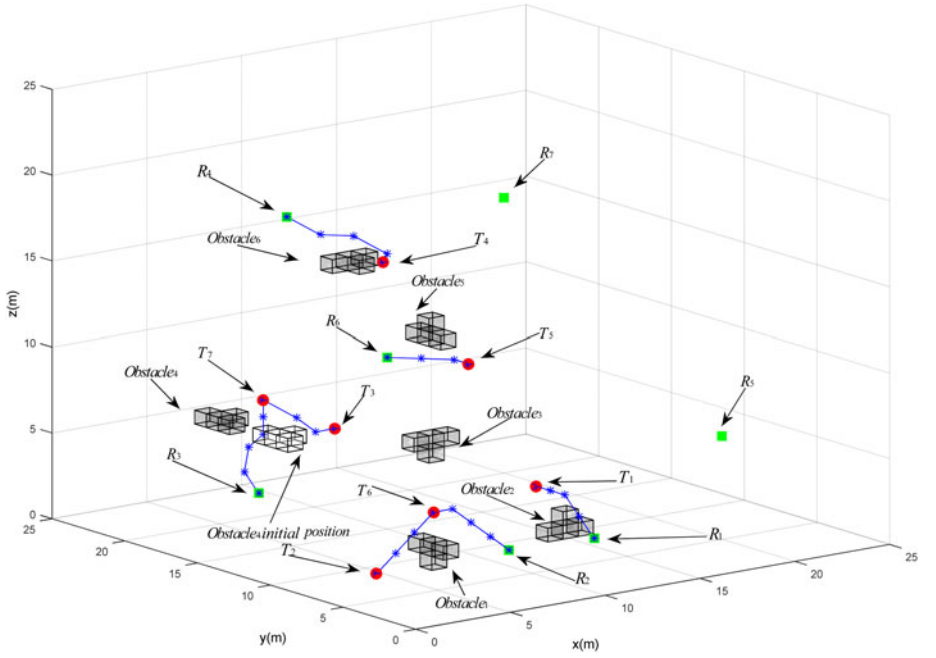


Figure 8. Task assignment and path planning in a dynamic environment.

5.2. *Task assignment and path planning in a 3D dynamic environment.* Obstacles can be divided into static obstacles and dynamic obstacles in the real ocean environment, and it is more complicated to deal with dynamic obstacles.

The simulation result of task assignment and path planning in a dynamic environment is shown in Figure 8. $Obstacle_4$ is a dynamic obstacle. Since $Obstacle_4$ is moving, the moving path of R_3 near $Obstacle_4$ is affected. The dynamic $Obstacle_4$ is far away from the other AUVs, so the impact of the other AUVs' path planning is not large. Since $Obstacle_4$ is relatively far away from R_3, the path planning of R_3 is basically not affected. With the moving of $Obstacle_4$ and R_3 getting closer to $Obstacle_4$, the movement of $Obstacle_4$ begins to affect the path planning of R_3. A path to steer clear of obstacles for R_3 is planned when $Obstacle_4$ is not moving. When $Obstacle_4$ is dynamic, Figure 8 shows that R_3 changes the path of the original selection to pass through the initial position of $Obstacle_4$.

6. CONCLUSION. A GBSOM algorithm is proposed to complete AUV task assignment and path planning in 3D static and dynamic environments in this paper. The algorithm can overcome the problem of speed jumps and offers obstacle avoidance capabilities missing from traditional SOM algorithms. Simulation results show that the GBSOM algorithm proposed in this paper can not only achieve the task assignment in a 3D underwater environment, but can also avoid static or dynamic obstacles and resolve the speed jump problem. In this paper, the task assignment and path planning are considered under ideal conditions with no ocean current. Ocean currents will be added in the next step of research.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press

ACKNOWLEDGMENT

This project is supported by the National Natural Science Foundation of China (51575336, 61503239), Creative Activity Plan for Science and Technology Commission of Shanghai (16550720200, 15550722400).

REFERENCES

Aluizio, A.F.R. and Santana, O.V. (2015). Self-Organizing Map with Time-Varying Structure to Plan and Control Artificial Locomotion. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 1594–1607.

Cao, X. and Zhu, D.Q. (2015). Multi-AUV Underwater Cooperative Search Algorithm Based on Biological Inspired Neurodynamics Model and Velocity Synthesis. *Journal of Navigation*, **68**, 1075–1087.

Glasius, R., Komoda, A. and Gielen, S. (1995). Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, **8**, 125–133.

Hopfield, J.J. and Tank, D.W. (1985). *"Neural" computation of decisions in optimization problems*. Springer-Verlag New York, Inc.

Huang, H., Zhu, D.Q. and Ding, F. (2014). Dynamic task assignment and path planning for multi-AUV system in variable ocean current environment. *Journal of Intelligent Robot Systems*, **74**, 999–1012.

Huang, H., Zhu, D.Q. and Yuan, F. (2012). Dynamic Task Assignment and Path Planning for Multi-AUV System in 2D Variable Ocean Current Environment. *The 24th Chinese Control and Decision Conference*, 23–25.

Huang, Z., Zhu, D.Q. and Sun, B. (2016). A multi-AUV cooperative hunting method in 3-D underwater environment with obstacle. *Engineering Applications of Artificial Intelligence*, **50**, 192–200.

Jaillet, L. and Porta, J.M. (2013). Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE Transactions on Robotics*, **29**, 105–117.

Javier, D.L., Dario, M. and Yadira, Q. (2015). Self-Organizing Techniques to Improve the Decentralized Multi-task Distribution in Multi-robot Systems. *Neurocomputing*, **163**, 47–55.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing,* **21**, 1–6.

Kulkarni, I.S. and Pompili, D. (2013). Task allocation networked autonomous underwater vehicles in critical missions. *IEEE Journal on Selected Areas in Communications*, **28**, 716–727.

Luo, C., Gao, J., Li, X., Mo, H. and Jiang., Q. (2014a). Sensor-based Autonomous Robot Navigation under Unknown Environments with Grid Map Representation. *IEEE Symposium Series on Computational Intelligence (SSCI), IEEE*, 98–104.

Luo, C., Gao, J., Murphey, Y.L., and Jan, G.E. (2014b). A Computationally Efficient Neural Dynamics Approach to Trajectory Planning of An Intelligent Vehicle. *IEEE International Joint Conference on Neural Networks (IJCNN), IEEE,* 934–939.

Luo, L.Z., Chakraborty, N. and Sycara, K. (2014). Provably-Good Distributed Algorithm for Constrained Multi-Robot Task Assignment for Grouped Tasks. *IEEE Transactions on Robotics*, **31**, 19–30.

Ren, S.X. and Mei, Y. (2013). Underwater glider task allocation based on the ant colony algorithm. *Ninth International Conference on Natural Computation*, 585–589.

Sahar, T., Anis, K., Omar, C., Habib, Y., Hachemi, B., Mohamed-Foued, S. and Yasir, J. (2014). A Distributed Market-based Algorithm for the Multi-robot Assignment Problem. *Procedia Computer Science*, **32**, 1108–1114.

Zhu, A. and Yang, S.X. (2006). A neural network approach to dynamic task assignment of multi-robots. *IEEE Transactions on Neural Network*s, **17**, 1278–1287.

Zhu, A. and Yang, S.X. (2012). An improved SOM-based approach to dynamic task assignment of multi-robots. *World Congress on Intelligent Control and Automation, Jinan, China, July,* 2168–2173.

Zhu, D.Q., Huang, H. and Yang, S.X. (2013). Dynamic Task Assignment and Path Planning of Multi-AUV System based on an Improved Self-Organizing Map and Velocity Synthesis Method in Three-Dimensional Underwater workspace. *IEEE Transactions on Cybernetics*, **43**, 504–514.

https://doi.org/10.1017/S0373463317000728 Published online by Cambridge University Press