

Multi-AUV Cooperative Target Search Algorithm in 3-D Underwater Workspace

Xiang Cao and A-long Yu

(School of Physics and Electronic Electrical Engineering, Huaiyin Normal University,
Huaiian 223300, China)
(E-mail: cxeffort@126.com)

To improve the efficiency of multiple Autonomous Underwater Vehicles (multi-AUV) cooperative target search in a Three-Dimensional (3D) underwater workspace, an integrated algorithm is proposed by combining a Self-Organising Map (SOM), neural network and Glasius Bioinspired Neural Network (GBNN). With this integrated algorithm, the 3D underwater workspace is first divided into subspaces dependent on the abilities of the AUV team members. After that, tasks are allocated to each subspace for an AUV by SOM. Finally, AUVs move to the assigned subspace in the shortest way and start their search task by GBNN. This integrated algorithm, by avoiding overlapping search paths and raising the coverage rate, can reduce energy consumption of the whole multi-AUV system. The simulation results show that the proposed algorithm is capable of guiding multi-AUV to achieve a multiple target search task with higher efficiency and adaptability compared with a more traditional bioinspired neural network algorithm.

KEY WORDS

1. Cooperative target search.
2. Multi-AUV. Self-organising map.
3. Glasius bioinspired neural network.

Submitted: 15 October 2016. Accepted: 16 May 17. First published online: 30 June 2017.

1. INTRODUCTION. Multiple Autonomous Underwater Vehicles (multi-AUV) target search means several AUV cooperate to seek and recognise a target in the underwater environment using their carried sensors. Target search is an important element in realising underwater rescue, detection, archaeology and underwater warfare. This is one of the key areas of underwater vehicle research (Woolsey and Techy, 2009; Paull et al., 2014; Matsuda et al., 2012). Hence, multi-AUV target search, which provides more possible solutions with their high parallelism, robustness and high efficiency collaboration have gradually attracted researchers' attention (Lapierre and Jouvencel, 2008; Lynch and Ellery, 2014; Fiorelli et al., 2006).

Multi-robot teams on land have been studied for decades, much has been achieved and many algorithms for search strategies have been proposed. For instance, Gabrieli and

Rimon (2003) proposed a spiral tree search algorithm and Gonzalez et al. (2005) put forward a backtracking spiral algorithm. However, these are low in efficiency because of their long search chain and demand on robots' starting points. Polycarpou et al. (2001) adopted a traversal exploration algorithm to make a continuous linear search. Although these trajectory-based methods realise cooperation in the simplest way, they are very low in efficiency and suitable only for searches in static environments. Another category of algorithms depends on real-time environment information to make appropriate adjustments in path planning. For example, the Hierarchical Reinforcement Learning (HRL) algorithm for an unknown environment was proposed by Cai et al. (2013). A Particle Swarm Optimisation (PSO) algorithm for path planning and tracking was proposed by Cai and Yang (2013). Various bioinspired neural network models have also been proposed (Yang and Meng, 2003; Ni and Yang, 2011; Luo and Yang, 2008) for optimising search paths. Without rigid trajectories set in advance, these approaches plan more reasonable paths according to real-time environmental information. Since these paths are collision-free and shorter, they are safer and time and energy-saving.

In comparison with multi-robot teams on land, multi-AUV team collaboration is much more difficult. As their energy supply is limited, they must finish the assignment as fast as possible. Furthermore, practical problems such as underwater communication and a Three-Dimensional (3D) working environment add more problems. Thus, studies on cooperative search by multi-AUV teams are lagging. Work in this field includes that by Cui et al. (2010) who adopted a leader-follower formation strategy which is designed to allow for weak communication exchange, low bandwidth and low update rates in the underwater environment. By this approach, fewer information exchanges among multi-AUV team members are needed compared to other algorithms, but the shortcoming is that some AUV members may not engage in the search task but are only involved in maintaining a formation. This may waste energy and contributes very little to efficient working.

Yoon and Qiao (2011) proposed a synchronisation-based survey to conduct a multi-AUV team search with limited energy and communication capabilities. This approach enables the work team to search large areas even with mechanical failures of some team members. However, for its centralised control and lawn-mowing-type search paths, it is also not high in search efficiency and mainly applicable to searches for static targets.

Couillard et al. (2012) developed a local sequential path planning algorithm combined with a global simulated annealing algorithm allowing a multi-AUV team to search for more than one target while minimising the total distance covered. However, this method is concerned only with static targets whose positions are already known in advance. Its exclusion of the effects of obstacles and ocean current made it inadequate for searches in practical use.

Cao and Zhu (2015) made an in-depth study in multi-AUV target search. They made a combination of a velocity synthesis algorithm with a biologically-inspired neurodynamics model for an underwater environment subjected to ocean currents. In this paper, an integrated algorithm for multi-AUV cooperative target search by combining a biologically-inspired neurodynamics model and velocity synthesis algorithm is proposed. It is expected to provide shorter paths than other algorithms in underwater environments with ocean current and obstacles. The biologically-inspired neurodynamics model algorithm is developed to coordinate AUV cooperation, and plans their search paths to avoid obstacles. A velocity synthesis algorithm is applied to make a shorter search path to offset the effect of ocean current. Effectiveness and applicability of the proposed integrated cooperative target search

method are proved by simulation. However, this method did not include task assignment, which reduces the searching efficiency.

In this paper, we study an algorithm for cooperative target search in a 3D underwater workspace. To improve cooperation efficiency and avoid overlapping coverage, a new strategy is proposed by combining a Self-Organising Map (SOM), neural network and Gladius Bioinspired Neural Network (GBNN). The whole workspace is divided into four subspaces and each AUV will be allocated a certain search task. Because of the similar characteristics between the multi-AUV system and a SOM neural network, the SOM algorithm can be used in multi-AUV task assignment. In cases where there is no prior knowledge of the dynamic environment, and without any learning procedures, GBNN is developed to plan AUVs' search paths. By combining the two algorithms, it is expected to avoid overlapping search paths and raise the coverage rate within the given workspace and thus reduce energy consumption of the whole system.

The advantages of this algorithm can be summarised as follows. Firstly, by dividing the workspace into subspaces, it balances each AUV's assignment and thus will not fail a search task due to energy exhaustion of some overburdened AUV members. By combining SOM & GBNN, it enhances cooperation within the multi-AUV system and thus lowers energy consumption and the amount of repetitive coverage to improve work efficiency. Finally, our algorithm can reduce the calculation burden.

The rest of the paper is organised as follows. In Section 2, the problem statement is given. Section 3 presents the proposed SOM & GBNN-based integrated search algorithm. Simulation experiments for various situations are related in Section 4. Section 5 discusses the sensitivity of the parameters of the proposed algorithm in detail. The conclusion is given in Section 6.

2. PROBLEM STATEMENT. This paper demonstrates the performance of the proposed algorithm for the Poseidon SB-1 submarine of the Taiwan LeiHu company, as shown in [Figure 1](#). The mechanical part of the underwater vehicle prototype is used, and its control system is redesigned. An attitude sensor is added, to meet the demand for control in this project. The main parameters of the underwater robot are: Displacement: 7.7 kg (floating on the surface of the water), 7.95 kg (descending); Total length: 774 mm; Total width: 290 mm; hull width: 200 mm; three blade propeller, diameter: 40 mm and section: 41 mm. The shell of the submarine is solid. Built-in pulse pumps and water bags can make the underwater robot move up and down. A gyroscope can detect attitude angle, and using the attitude sensor, real-time data is processed.

A multi-AUV system is designed to search for targets in a 3D underwater environment. In a typical search space, there are AUVs, targets and several obstacles of different size and shape ([Figure 2](#)). The turning radius of the AUV is trivial compared with the dimensions of the workspace, thus the AUV is assumed to be able to move omnidirectionally. AUVs are assumed to understand nothing about the underwater environment but the total numbers of targets and the boundaries of their workspace. AUVs are also assumed to be able to distinguish themselves from their targets. AUVs and targets are only allowed to move within the search space. The targets move at random, while AUVs move according to the proposed algorithm. When any target moves into any AUV's sensing range, this target is regarded as being found. After all targets within the given workspace have been found, the search task comes to an end.



Figure 1. Poseidon SB-1 experiment platform.

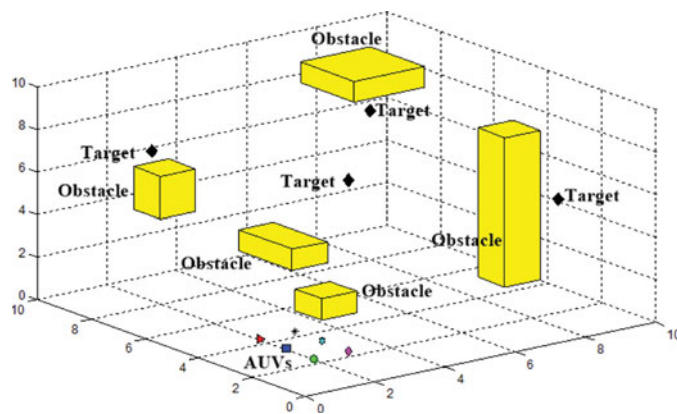


Figure 2. A diagram of the search task.

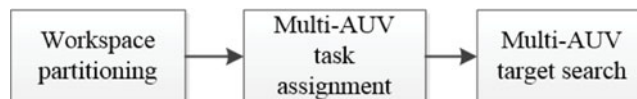


Figure 3. Operational flow of the multi-AUV target search.

3. SEARCH ALGORITHM. The operational flow of the SOM & GBNN algorithm is shown in Figure 3. For multi-AUV cooperative target search in a 3D underwater workspace, the overall task involves three phases. First, concerning a balanced assignment for each AUV, the 3D workspace is divided into four average subspaces S_1 , S_2 , S_3 and S_4 (shown in Figure 4). Then, tasks will be assigned according to the SOM algorithm which ensures one AUV for each subspace. After finishing the task assignment, AUVs will move to the assigned subspace in the shortest manner. Lastly, the GBNN is developed to guide the AUVs to achieve target search in the subspaces. The combination of the SOM & GBNN algorithm operates without explicitly searching over the free space or the collision paths and without explicitly optimising any global cost functions. Furthermore, it requires fewer function evaluations and parameter adjustments.

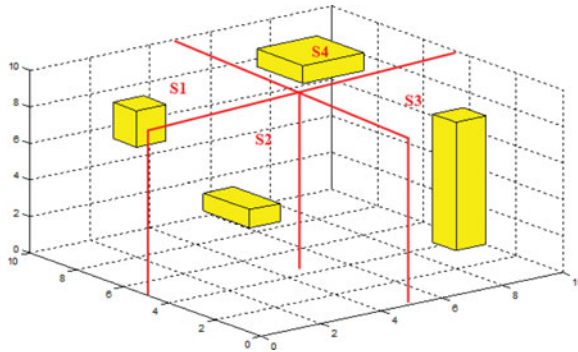


Figure 4. Workspace partitioning.

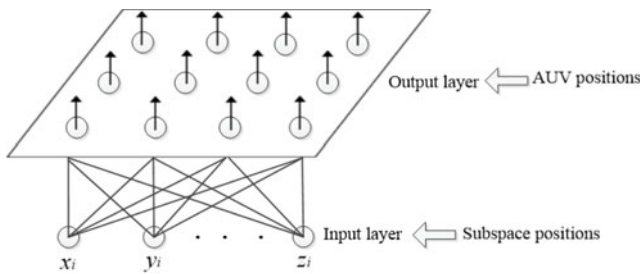


Figure 5. Structure of the SOM neural network.

3.1. *Multi-AUV task assignment.* The SOM neural network model that contains two layers is presented in Figure 5. When the SOM neural network is used in multi-AUV task assignment, the first layer is the input layer including Q neurons which represent the Cartesian coordinates of the subspace; three parameters (x_i, y_i, z_i) form one input neuron. The second layer is the output layer that involves the AUV’s coordinates and corresponding paths. Each neuron of the output layer connects to the neurons in the input layer. The strength of each output neuron is given by a 3D weight vector, which is initialised as the initial AUV’s coordinate. Subspaces are input into the network one by one until the last subspace is input. The iterations continue until all subspaces have been assigned an AUV (Zhu et al., 2013).

We divide the overall problem into sub-problems including the rule to select, for the winner, the definition of the neighbourhood function and the rule to update weights. First, the SOM algorithm is applied to determine which AUVs are the winners for subspaces. Then, the neighbourhood function determines which AUVs are the neighbours of the corresponding winners and, meanwhile, figures out the moving speed for winners as well as neighbours. Finally, it guarantees an AUV reaches its subspace along the shortest path (Cao and Zhu, 2015; Kohonen, 1982).

3.1.1. *Winner selection rules.* The winner is determined by the following expression (Zhu and Yang, 2006):

$$[P_j] \Leftarrow \min \{D_{ijh}, i = 1, 2, \dots, I; j = 1, 2, \dots, J; h = 1, 2, \dots, H\} \quad (1)$$

where $[P_j]$ denotes that the j -th neuron from the i -th group is the selected winner to the h -th input neuron. D_{ijh} is a value related to Euclid distance between the two correlated neurons. Selecting the winner depends on how to define and calculate D_{ijh} during iterations. First, an equation is given to interpret the Euclid distance between two neurons (Hendzel, 2005):

$$D_{ijh} = |T_i - R_{jh}| = \sqrt{(x_i - w_{jhx})^2 + (y_i - w_{jhy})^2 + (z_i - w_{jhz})^2} \quad (2)$$

$T_i = (x_i, y_i, z_i)$ is the coordinate of the i -th input neuron in the 3D coordinate system, which also denotes the location of the subspaces. $R_{jh} = (w_{jhx}, w_{jhy}, w_{jhz})$ is the coordinate of the output neurons, representing the location of a certain AUV at a certain time.

3.1.2. *Neighbourhood updating rules.* After the winner is selected, the next step is to design the neighbourhood function and compute weights of the winner and its neighbours. The neighbourhood is designed as a sphere with radius λ where the centre is the winner node. The neighbourhood function determines the attractive strength of the input data on the winner as well as its neighbours. The influence on selected neurons diminishes as the distances between the neighbour neuron and the winner increase. The neighbours of the winner are determined according to the following equation (Reeve and Hallam, 2005):

$$f(d_m, g) = \begin{cases} e^{-d_m^2/g^2(t)}, & d_m < \lambda \\ 0, & d_m \geq \lambda \end{cases} \quad (3)$$

$d_m = |N_m - N_j|$ is the distance between the m -th output neuron and the winner. λ is set to a constant value denoting neighbourhood range. The function $g^2(t) = (1 - \partial)^t g_0$ is nonlinear, where t represents the number of iterations. ∂ is the update rate determining the calculation time, and $\partial < 1$ is constant. The computation time diminishes as the parameter ∂ increases.

The next step is to move the winners as well as the corresponding neighbours along optimum paths. The update rule is defined as (Miyata et al., 2002):

$$R_{jh}(t+1) = \begin{cases} T_j & D_{ijh} \leq D_{min} \\ R_{jh}(t) + \partial \times f(d_m, g) \times (T_i - R_{jh}(t)), & D_{ijh} > D_{min} \end{cases} \quad (4)$$

D_{min} is the termination condition of iterations. As long as $D_{ijh} \leq D_{min}$, the weight will be replaced by the corresponding target coordinate. Obviously, the weight's amendment is not only decided by the winner and neighbour's locations but also decided by the neighbourhood function and network learning speed ∂ .

As shown in Figure 6, the 3D underwater workspace where four AUVs R1, R2, R3 and R4 are ready for task assignment is divided into four equal subspaces S1, S2, S3 and S4. According to the principle of SOM, the four subspaces should be taken as input neuron of the SOM neural network and AUVs' positions as the output neuron, each input should correspond to an output, i.e. each subspace should be allocated one AUV. By the competition rules in Equations (1) and (2), R1, R2, R3 and R4 are allocated respectively to S2, S3, S1 and S4. Once task assignment is settled, each AUV will get to its subspace along the shortest path.

3.2. *Multi-AUV target search.* In the 3D underwater workspace, the target search is determined by the GBNN. The GBNN algorithm in this paper is a kind of discrete Hopfield-type neural network. Firstly, the Glasius bioinspired neural network is built according to the underwater workspace (as in Figure 7). In this model, each individual neuron is connected

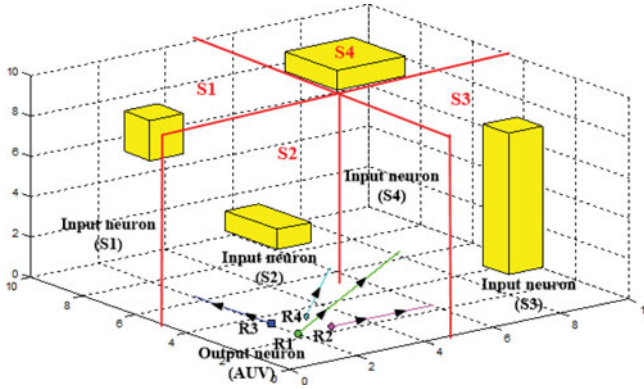


Figure 6. Process of multi-AUV task assignment.

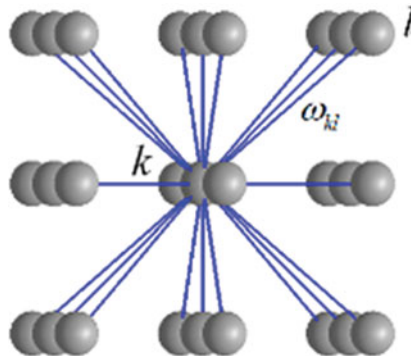


Figure 7. A diagram of the neural network.

with the adjacent ones to form a network for transmission of their activity. In the proposed model, the excitatory input results from the target and the lateral neural connections, while the inhibitory input results from the obstacles only. Each neuron has local lateral connections to its neighbouring neurons. The neuron responds only to the stimulus within its receptive field (Cao et al., 2016). In the proposed model, collision-free AUV motion is planned in real time based on the dynamic activity landscape of the neural network. The dynamics of this discrete time neural network are described in the following equations (Luo et al., 2014a).

$$x_k(t + 1) = g \left(\sum_{l \in S_l} \omega_{kl} x_l(t) + I_k \right) \tag{5}$$

$$\omega_{kl} = \begin{cases} e^{-\gamma|k-l|^2}, & |k - l| \leq r \\ 0, & |k - l| > r \end{cases} \tag{6}$$

where ω_{kl} are symmetric connection weights between the k -th neuron and the l -th neuron; $|k - l|$ is the Euclidian distance from the k -th neuron to the l -th neuron; $g(\cdot)$ is the transfer

function and γ and $\gamma > 0$ are constants. The external input I_k to the k -th neuron is defined as $I_k = E$, if it is a target; $I_k = -E$, if it is an obstacle position; $I_k = 0$, otherwise, where $E \gg 1$ it is a positive constant.

$$I_i = \begin{cases} E, & \text{if it is a target} \\ -E, & \text{if it is an obstacle} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Transfer function $g(\cdot)$ may be any monotonically increasing function (Glasius et al., 1995; 1996). A piecewise linear function is selected as the transfer function as follows.

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

where $\beta > 0$ is a positive constant.

The proposed network characterised by Equation (5) guarantees that the positive neural activity can be propagated to all the state space, but the negative activity only stays locally. Therefore, the target globally attracts the AUV, while the obstacles only locally avoid the collision. The activity landscape of the neural network dynamically changes due to the varying external inputs from the targets, obstacles and the internal activity propagation among neurons. The optimal AUV path is planned from the dynamic activity landscape, and the previous AUV location. The AUV will move to the neuron with maximal neural activity. After it reaches its next location, the next location becomes a new current location. The current AUV location adaptively changes according to the varying environment. The activity landscape of the neural network dynamically changes due to the varying external inputs from the target and obstacles and the internal activity propagation among neurons. For energy and time efficiency, the robot should travel the shortest path and minimise turning. For a given current robot location, denoted by L_c , the next robot location L_n is obtained by (Luo et al., 2014b)

$$L_n = \arg \max_{m,n} (x(m, n)) \in \{N_s | (m, n)\} \quad (9)$$

where s is the number of neighbouring neurons of the L_c -th neuron ($s = 26$), i.e. all the possible next locations of the current location L_c . Variable $x(m, n)$ is the neural activity of the l -th neuron.

In the search task, the neural activity landscape will never reach a steady state in 3D environments. The AUV keeps moving toward the neuron location with the maximum activity in the AUV detection region. In the proposed model, due to the very large external input constant E , the target and obstacles stay at the peak and valley of the activity landscape of the neural network, respectively. Thus, the AUV should be able to search for the target efficiently with obstacle avoidance until the search task ends. In this way, the AUVs can realise a cooperative search efficiently and naturally. This is the main difference between the proposed algorithm and other algorithms for cooperative search.

4. SIMULATION STUDIES. To demonstrate the effectiveness of the proposed algorithm, several different cases, including searches for static and dynamic targets and

Table 1. Control parameters.

Parameters	ϑ	γ	β	r	E
Value	0.8	2	0.01	$\sqrt{3}$	100

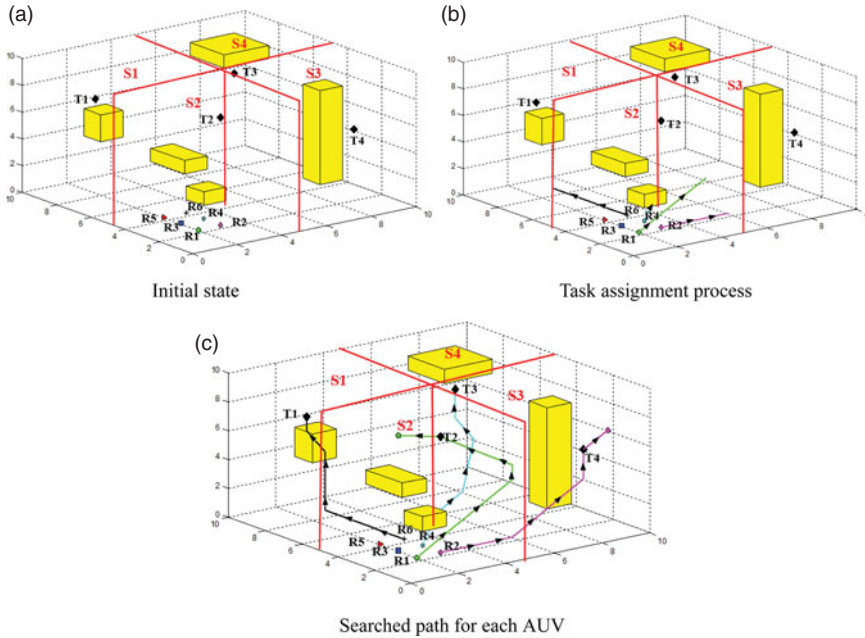


Figure 8. Search process with static targets based on SOM & GBNN algorithm.

when some AUV members break down, were implemented on the software platform MATLAB R2011a. Simulations were carried out with different models of 3D underwater workspaces with targets randomly distributed. In all simulation cases, AUVs were put into a workspace of $10 \times 10 \times 10$. It is assumed that AUV members are not informed in advance of their working environment except the total numbers of targets and the boundaries of the underwater workspace. The sensing range of each AUV is set as $\sqrt{3}$ and the energy consumption for each step taken is 1. The parameters of the proposed algorithm in all the experiments are listed in Table 1.

4.1. *Search for static targets.* In the case of a search for static targets, obstacles and static targets are randomly distributed in the workspace (shown in Figure 8(a)). The initial coordinates of the four static targets are respectively (8,8,7), (1,7,8), (8,1,6) and (2,1,9), and the initial coordinates of the six AUVs are (1,1,1), (2,1,1), (1,2,1), (2,2,1), (1,3,1) and (2,3,1). According to the proposed algorithm, the workspace is equally divided into four subspaces S1, S2, S3 and S4. After that, through calculation from Equations (1)–(4) by SOM, winners R1, R2, R4 and R6 are assigned respectively to S2, S3, S4 and S1 (shown in Figure 8(b)). Since R3 and R5 fail in the competition, they will not join in the search task but remain stationary. After each AUV gets to the subspace it takes charge and GBNN automatically plans a collision-free search path for each searcher according to Equations (5)–(9). As shown in Figure 8(c), targets T1, T2, T3 and T4 are found by AUVs R6, R1,

Table 2. Motion coordinates of the six AUVs based on SOM & GBNN.

AUV	R1	R2	R3	R4	R5	R6
Initial location	(1,1,1)	(2,1,1)	(1,2,1)	(2,2,1)	(1,3,1)	(2,3,1)
1 step	(2,1,2)	(3,1,1)	–	(3,3,1)	–	(2,4,1)
2 step	(3,1,3)	(4,1,1)	–	(4,4,1)	–	(2,5,1)
3 step	(4,1,4)	(5,1,1)	–	(5,5,1)	–	(1,6,2)
4 step	(5,1,5)	(6,1,2)	–	(6,5,2)	–	(1,6,3)
5 step	(5,1,6)	(7,1,3)	–	(7,6,3)	–	(1,6,4)
6 step	(4,1,7)	(8,1,4)	–	(8,7,4)	–	(1,6,5)
7 step	(3,1,8)	(8,1,5)	–	(8,8,5)	–	(1,6,6)
8 step	(2,1,9)	(8,1,6)	–	(8,8,6)	–	(1,7,7)
9 step	(1,2,9)	(9,1,7)	–	(8,8,7)	–	(1,7,8)
10 step	(2,3,9)	(10,1,6)	–	(9,9,7)	–	(2,7,9)

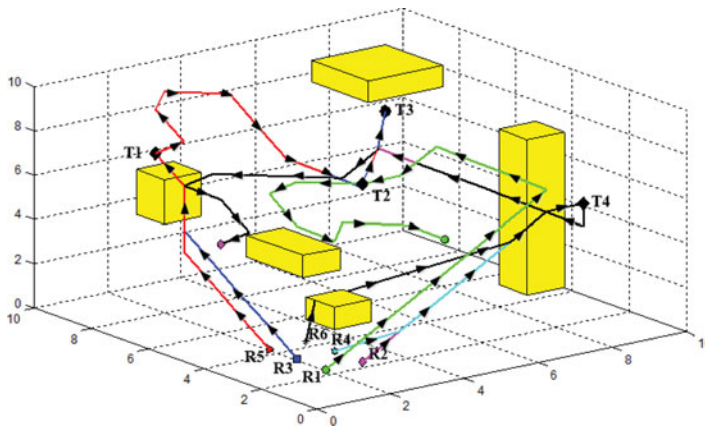


Figure 9. Search process with static targets based on BNN algorithm.

R4 and R2 respectively. Table 2 gives the motion coordinates of the six AUVs during the search process. From the list, we find that the four AUVs who have been assigned with the search task move only ten steps to find the targets, while the other two AUVs remain stationary. This proves that the proposed algorithm not only works with high efficiency but also saves energy for the whole work team.

In contrast, Figure 9 and Table 3 present the search results for multiple static targets by only a Bioinspired Neural Network (BNN). Under the same circumstances, BNN, which does not make a workspace partition as well as task assignment, put six AUVs altogether into the search task at the same time and thus each AUV had to take 19 steps before all four targets were detected. By comparison, we find that the simple use of only BNN is much lower in work efficiency than an integrated use of SOM & GBNN.

As assumed, each step taken consumes one unit of the energy. By BNN, each of the six AUVs takes 19 steps, so the total energy consumption should be 114. In comparison, SOM & GBNN takes only 40, which is much lower than the BNN algorithm. Through the analysis above, we can see that the introduction of SOM for workspace partition ensures a more efficient allocation of search tasks and thus conserves energy for the whole work system. In Table 4 and Figure 10, energy consumption of each step taken by AUVs as a

Table 3. Motion coordinates of the six AUVs based on BNN.

AUV	R1	R2	R3	R4	R5	R6
Initial location	(1,1,1)	(2,1,1)	(1,2,1)	(2,2,1)	(1,3,1)	(2,3,1)
1 step	(2,1,2)	(3,1,2)	(1,3,2)	(3,1,2)	(1,4,2)	(3,4,2)
2 step	(3,1,3)	(4,1,3)	(1,4,3)	(4,1,3)	(1,5,3)	(4,3,3)
3 step	(4,1,4)	(5,1,4)	(1,5,4)	(5,1,4)	(1,6,4)	(5,2,4)
4 step	(5,1,5)	(6,1,5)	(1,6,5)	(6,1,5)	(1,6,5)	(6,1,5)
5 step	(6,1,6)	(7,1,6)	(1,6,6)	(7,1,6)	(1,6,6)	(7,1,6)
6 step	(7,1,7)	(8,1,6)	(1,6,7)	(8,1,6)	(1,6,7)	(8,1,6)
7 step	(6,1,8)	(8,1,5)	(1,7,8)	(8,1,5)	(1,7,8)	(8,1,5)
8 step	(5,1,9)	(7,2,6)	(1,6,9)	(7,2,6)	(1,6,9)	(7,2,6)
9 step	(4,1,10)	(6,2,7)	(1,7,10)	(6,2,7)	(1,7,10)	(6,2,7)
10 step	(3,1,9)	(5,3,8)	(2,8,10)	(5,3,8)	(2,8,10)	(5,3,8)
11 step	(2,1,9)	(4,3,9)	(3,7,10)	(4,3,9)	(3,7,10)	(4,3,9)
12 step	(1,2,9)	(3,3,8)	(3,6,9)	(3,3,8)	(3,6,9)	(3,3,8)
13 step	(1,3,8)	(2,4,8)	(3,5,8)	(2,4,8)	(3,5,8)	(2,4,8)
14 step	(1,2,7)	(1,5,8)	(4,5,7)	(1,5,8)	(4,5,7)	(1,5,8)
15 step	(2,2,6)	(1,6,7)	(5,5,6)	(1,6,7)	(5,5,6)	(1,6,7)
16 step	(3,3,6)	(2,6,6)	(6,6,6)	(2,6,6)	(6,6,6)	(2,6,6)
17 step	(4,2,6)	(2,5,5)	(7,7,6)	(2,5,5)	(7,7,6)	(2,5,5)
18 step	(5,2,5)	(2,6,4)	(8,8,7)	(2,6,4)	(8,8,7)	(2,6,4)
19 step	(5,3,4)	(2,7,3)	(9,8,6)	(2,7,3)	(9,8,6)	(2,7,3)

Table 4. Energy consumption with two different algorithms.

Step										
Algorithm	1	2	3	4	5	6	7	8	9	10
SOM & GBNN	4	8	12	16	20	24	28	32	36	40
BNN	6	12	18	24	30	36	42	48	54	60
Step	11	12	13	14	15	16	17	18	19	
Algorithm										
SOM & GBNN	–	–	–	–	–	–	–	–	–	–
BNN	66	72	78	84	90	96	102	108	114	

work team is given. From the statistics presented, we find that by BNN algorithm AUVs not only consume more energy as a whole work team, but also in each step in the search process.

Regarding search overlapping rate, Table 5 and Figure 11 give a comparison of the two algorithms. Since SOM & GBNN makes a partition of the workspace for proper allocation of AUV searchers, it is free of overlapping search spaces during the search process except at the initial stage when several AUVs are close to each other. For BNN, however, several AUVs make overlapping searches in the same spaces and the overlapping rate even gradually increased with the progress of the search task. From Figure 11, we can also find overlapping search paths by different AUVs. Hence through the comparison of overlapping rate, it proves that the proposed algorithm works better at promoting cooperation between the AUVs.

We can also examine the environmental coverage rate, which means how much the searched spaces account for the whole workspace. Under the same circumstances, the environmental coverage rate increases and the work efficiency of the multi-AUV system

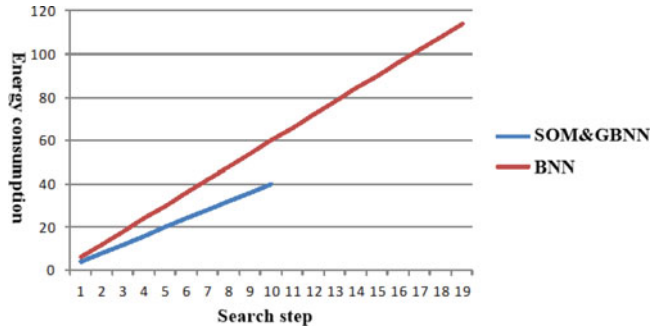


Figure 10. Energy consumption with two different algorithms.

Table 5. Overlapping rate with two different algorithms.

Step	1	2	3	4	5	6	7	8	9	10
Algorithm										
SOM & GBNN	1%	1%	1%	1%	1%	1%	1%	1%	1%	1%
BNN	1%	1.75%	2.63%	5.26%	7.89%	10.5%	13.1%	15.8%	18.4%	21.9%
Step	11	12	13	14	15	16	17	18	19	
Algorithm										
SOM & GBNN	—	—	—	—	—	—	—	—	—	—
BNN	23.7%	26.3%	28.9%	32.5%	34.2%	36.0%	39.5%	42.1%	43.9%	

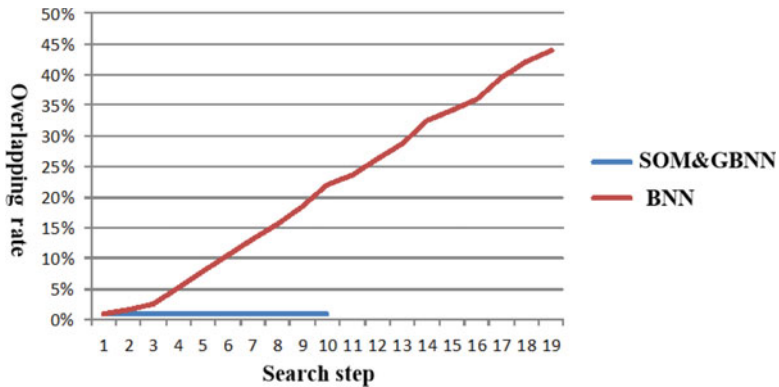


Figure 11. Overlapping rate with two different algorithms.

improves. In Table 6 and Figure 12, coverage rates by the two different algorithms are given. We can see that BNN has a higher total coverage rate because it involves two more AUVs and takes many more steps before the targets are detected. But from the eighth step, SOM & GBNN begins to outpace BNN. This proves SOM & GBNN also has a strong ability to explore the working environment.

4.2. *Search for dynamic targets.* The proposed algorithm not only applies in searching for static targets, but also for dynamic ones. For dynamic targets, a big challenge is whether the AUVs can adjust themselves according to information newly input during the search process. Since the GBNN algorithm is characterised by a dynamic path-planning

Table 6. Coverage rate with two different algorithms.

Step	1	2	3	4	5	6	7	8	9	10
SOM & GBNN	10.8%	15.6%	21.4%	27.2%	32.4%	37.6%	42.8%	48.6%	54.0%	60.0%
BNN	13.5%	21.4%	26.8%	30.8%	35.7%	39.2%	43.9%	47.5%	52.3%	56.8%
Step	11	12	13	14	15	16	17	18	19	
Algorithm										
SOM & GBNN	—	—	—	—	—	—	—	—	—	—
BNN	60.7%	63.2%	66.1%	69.4%	71.8%	73.1%	74.2%	75.8%	77.2%	

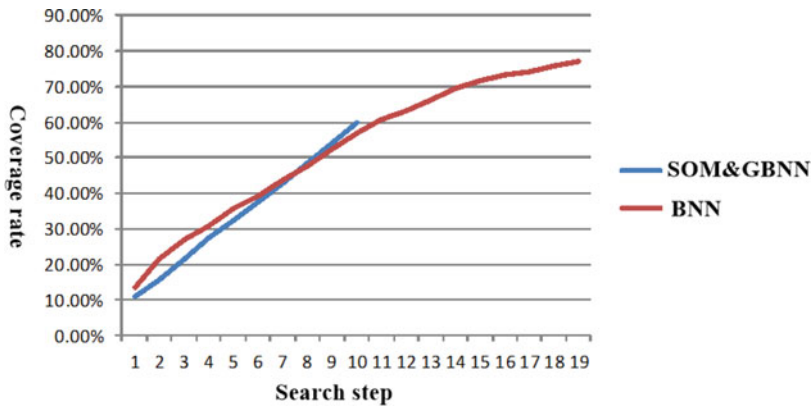


Figure 12. Coverage rate with two different algorithms.

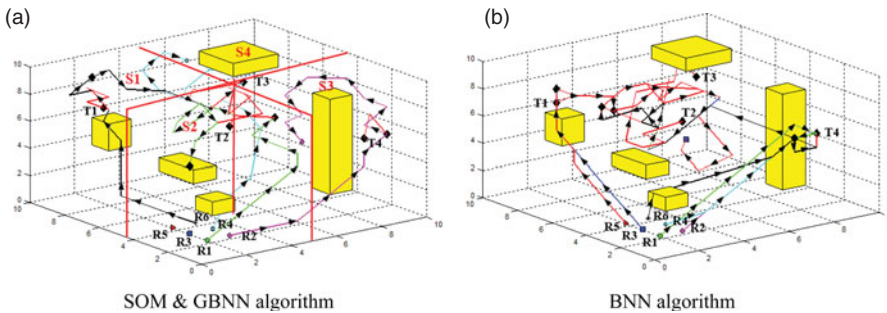


Figure 13. Search process with dynamic targets with two different algorithms.

mechanism according to changes of neurons' activity value generated by the targets' position, the AUVs can constantly renew their search paths to ensure a successful cooperative search. With the same initial starting points for AUVs and targets to that of the case of searching for static targets, Figure 13 gives the simulation searches for four dynamic targets which move at random in the whole workspace by BNN and the integrated algorithm of SOM & GBNN. By making a comparison, we find that SOM & GBNN works with higher efficiency and the AUVs cooperate better with each other. Moreover, its search paths overlap less.

Table 7. Performance comparison with two different algorithms.

Performance Algorithm	Search step	Overlapping rate	Energy consumption	Coverage rate
SOM & GBNN	15	3%	60	81.2%
BNN	21	49.5%	126	76.8%

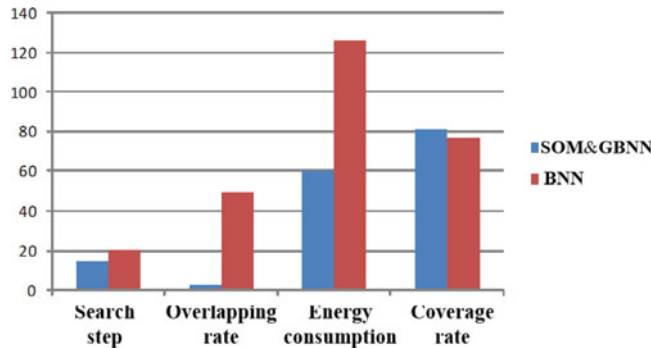


Figure 14. Performance comparison with two different algorithms.

To make a clear distinction between the two algorithms, Table 7 and Figure 14 list statistics of search steps, overlapping rate and energy consumption as well as the coverage rate by both algorithms. It can be concluded that the integrated algorithm of SOM & GBNN performs better than BNN in each item of simulation results. Hence it distinguishes itself with higher search efficiency and coverage rate, and at the same time lower overlapping rate and energy consumption.

In underwater workspaces, there are targets of random motion, but there are also targets with a predefined moving path. In this section, a simulation experiment of searching for targets with a predefined moving path is given. We assume that all targets have vertical downward movement. Figure 15 gives the simulation searches for four dynamic targets which move in a predefined path in the whole workspace. The result shows that all targets are found by the AUVs. Through the simulation experiment, this shows the proposed algorithm realises searches for targets of any movement.

4.3. *Some AUVs break down.* When searching in real 3D underwater workspaces, it is likely that one or more AUV will break down due to mechanical problems. In this event, it is an important index for measuring the proposed algorithm's cooperation to see whether the multi-AUV work system can complete its search task through internal adjustment. In this case, the simulation in this section deals with AUV failures. In the same simulation environment as that in the foregoing two sections, there are six AUVs involved in a search task for four static targets. At the beginning, only four AUVs R1, R2, R4 and R6 are assigned to S2, S1, S3 and S4. The other two AUVs R3 and R5 are left in an idle state. After a period, R6 breaks down and there is no AUV searcher in subspace S1 (shown as in Figure 16(a)). The SOM algorithm can assign tasks in a dynamic way when any mechanical failure occurs. According to Equations (1)–(4), the algorithm assigns an idle AUV to take over the work of the broken AUV. As shown in Figure 16(b), since R5 is closer to subspace S1 compared with R3, R5 should win the competition and be sent to

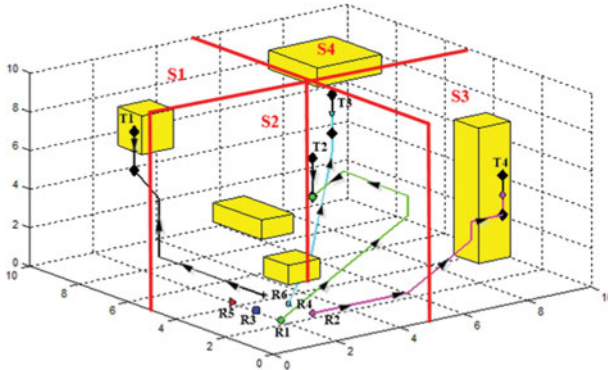


Figure 15. The performance with predefined target moving path.

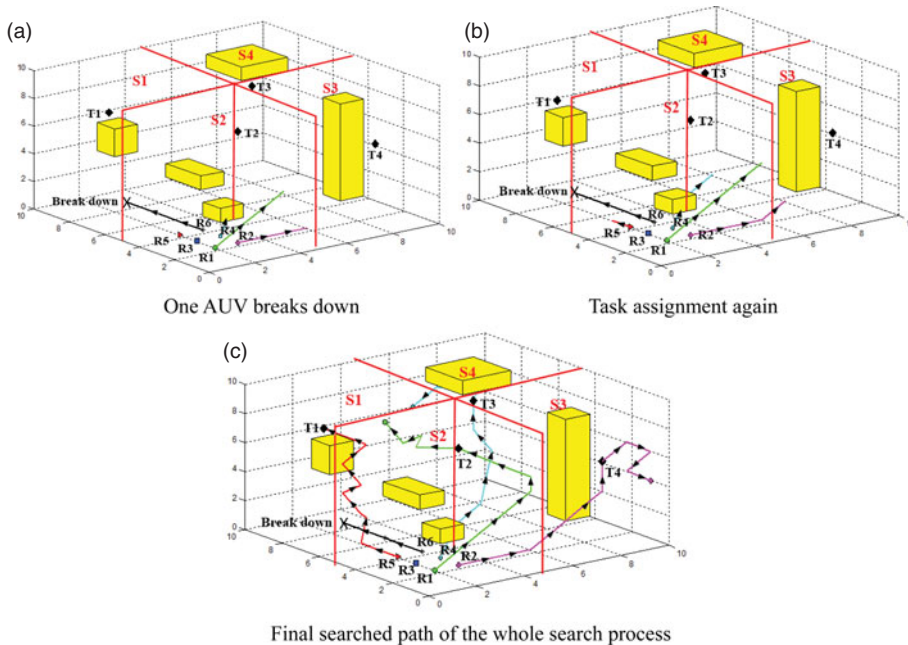


Figure 16. Search process when one of the team members breaks down.

subspace S1 to replace R6 for its search task. As shown in Figure 16(c), after R5 arrives at subspace S1, it continues with the search task and eventually detects target T1. From this simulation, it proves SOM & GBNN has the ability to complete a search task in the case of AUV mechanical failures through dynamic allocation. This also demonstrates the excellent cooperation of the proposed algorithm.

4.4. *In a large scale underwater environment.* To further discuss the performance of the proposed algorithm, a simulation experiment of a large scale underwater environment with larger sensor range is proposed. In the simulation case, AUVs are put into a workspace of $100 \times 100 \times 100$. It is assumed that the sensing range of each AUV is set as $10\sqrt{3}$. As

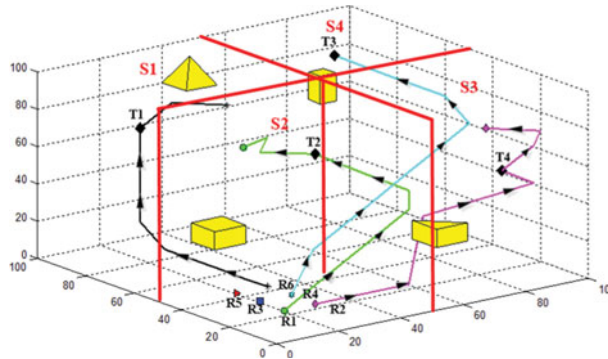


Figure 17. Search process with large scale underwater environment.

Table 8. Average time to achieve the search task by different parameter ϑ .

The value of parameter ϑ	The average time (s)
$\vartheta = 0.1$	38.2
$\vartheta = 0.3$	36.4
$\vartheta = 0.5$	35.5
$\vartheta = 0.7$	34.7

shown in Figure 17, targets T1, T2, T3 and T4 are found by AUVs R6, R4, R1 and R2 respectively. So the proposed algorithm not only applies to searches with a small sensor range, but also for a larger scale underwater environments with larger sensor range.

5. DISCUSSION. The results of the simulation experiments in Section 4 show that the proposed algorithm can satisfy the cooperative target search by multi-AUV under various conditions. The parameter sensitivity of the proposed algorithm is discussed in this section. The proposed algorithm is an integrated method aimed at real-world applications; most parameters of the proposed algorithm are decided by real-world applications, such as the velocity of AUVs and the targets, and the sensor range of the AUV. The most important parameters that need to be set by a designer come from the neural network in the proposed algorithm. The external input constant E is not an important factor, because the real-time AUV motion is generated by the relative value of the neural activities. So here only parameters ϑ , γ and β are discussed. To analyse the influence of parameters ϑ , γ and β , some simulations are carried out under the same conditions as the first experiment in Section 4, but with different values of parameters ϑ , γ and β . Because of the randomness of some factors, every simulation is conducted ten times. The average time to achieve the search task of the multi-AUV team at $\vartheta = 0.1$, $\vartheta = 0.3$, $\vartheta = 0.5$ and $\vartheta = 0.7$ is listed in Table 8. The average time to achieve the search task of the multi-AUV team at $\gamma = 3$, $\gamma = 5$, $\gamma = 8$ and $\gamma = 10$ is listed in Table 9. The average time to achieve the search task of the multi-AUV team at $\beta = 0.05$, $\beta = 0.1$, $\beta = 0.3$ and $\beta = 0.5$ is listed in Table 10.

The results in Tables 8, 9 and 10 show that the proposed algorithm is not very sensitive to the variations of the model parameters, so the parameters can be chosen from a very

Table 9. Average time to achieve the search task by different parameter γ .

The value of parameter γ	The average time (s)
$\gamma = 3$	37.9
$\gamma = 5$	35.8
$\gamma = 8$	34.6
$\gamma = 10$	36.3

Table 10. Average time to achieve the search task by different parameter β .

The value of parameter β	The average time (s)
$\beta = 0.05$	35.9
$\beta = 0.1$	37.1
$\beta = 0.3$	36.6
$\beta = 0.5$	35.1

wide range. All the case studies in this paper use the same model parameters, which are listed in Table 1.

6. CONCLUSION. In this paper, an integrated algorithm of self-organising map neural network and Glasius bioinspired neural network is introduced to deal with cooperative search by a multi-AUV team for static or dynamic targets. The SOM neural network can be used in multi-AUV task assignment, which strengthens cooperation among multi-AUV team members. It also makes full use of advantages of GBNN, i.e. without any previously known information about the workspace, without any pre-learning and with few parameter adjustments. Through simulation experiments with targets of different states, we have demonstrated that the proposed algorithm is able to plan shorter search paths to enhance work efficiency and save energy. For strong cooperation among AUV team members, the SOM & GBNN algorithm improves the coverage rate of the workspace, reduces overlapping rate, and can continue with the search work with some AUVs breaking down. Despite these advantages, there are still practical problems to be solved. For example, how should AUVs overcome the effects of ocean currents in underwater environments during their search process? Or how to figure out a more reasonable way of workspace partition and further improve the practicality of the algorithm.

ACKNOWLEDGEMENT

This project is supported by the University Science Research Project of Jiangsu Province (Major Program) (16KJA460003).

REFERENCES

- Cai, Y.F. and Yang, S.X. (2013). An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments. *International Journal of Control*, **86**, 1720–1732.

- Cai, Y.F., Yang, S.X. and Xu, X. (2013). A hierarchical reinforcement learning based approach to multi-robot cooperation for target searching in unknown environments. *Control and Intelligent Systems*, **41**, 1–12.
- 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.
- Cao, X., Zhu, D.Q. and Yang, S.X. (2016). Multi-AUV cooperative target search based on biological inspired neurodynamics model in 3-D underwater environments. *IEEE Transactions on Neural Networks and Learning Systems*, **27**, 1–11.
- Cao, X. and Zhu, D.Q. (2017). Multi-AUV task assignment and path planning with ocean current based on biological inspired self-organizing map and velocity synthesis algorithm. *Intelligent Automation & Soft Computing*, **23**, 31–39.
- Couillard, M., Fawcett, J. and Davison, M. (2012). Optimizing constrained search patterns for remote mine-hunting vehicles. *IEEE Journal of Oceanic Engineering*, **37**, 75–84.
- Cui, R.X., Ge, S.S., How, B.V.E. and Choo, Y.S. (2010). Leader-follower formation control of underactuated autonomous underwater vehicles. *Ocean Engineering*, **37**, 1491–1502.
- Fiorelli, E., Leonard, N., Bhatta, P., Paley, D., Bachmayer, R. and Fratantoni, D. (2006). Multi-AUV control and adaptive sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, **31**, 935–948.
- Gabriely, Y. and Rimon, E. (2003). Competitive on-line coverage of grid environments by a mobile robot. *Computational Geometry*, **24**, 197–224.
- Glasius, R., Komoda, A. and Gielen, S. (1995). Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, **8**, 125–133.
- Glasius, R., Komoda, A. and Gielen, S. (1996). A biologically inspired neural net for trajectory formation and obstacle avoidance. *Biological Cybernetics*, **74**, 511–520.
- Gonzlez, E., Alvarez, O. and Diaz, Y. (2005). A complete coverage algorithm, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2040–2044.
- Hendzel, Z. (2005). Collision free path planning and control of wheeled mobile robot using self-organizing map. *Technical Sciences*, **53**, 39–47.
- Kohonen, T. (1982). Analysis of a simple self-organizing process. *Biological Cybernetics*, **44**, 135–140.
- Lapierre, L. and Jouvencel, B. (2008). Robust nonlinear path-following control of an AUV. *IEEE Journal of Oceanic Engineering*, **33**, 89–102.
- Luo, C.M. and Yang, S.X. (2008). A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks*, **19**, 1279–1298.
- Luo, C.M., Gao, J.Y., Li, X.D., Mo, H.W. and Jiang, Q.M. (2014b). Sensor-based autonomous robot navigation under unknown environments with grid map representation. *IEEE Symposium on Swarm Intelligence*, 1–7.
- Luo, C.M., Gao, J.Y., Murphey, Y.L. and Jan, G.E. (2014a). A computationally efficient neural dynamics approach to trajectory planning of an intelligent vehicle. *International Joint Conference on Neural Networks*, Beijing, China, 934–939.
- Lynch, B. and Ellery, A. (2014). Efficient control of an AUV-manipulator system: an application for the exploration of Europa. *IEEE Journal of Oceanic Engineering*, **39**, 552–570.
- Matsuda, T., Maki, T. and Sakamaki, T. (2012). Performance analysis on a navigation method of multiple AUVs for wide area survey. *Marine Technology Society Journal*, **46**, 45–55.
- Miyata, N., Ota, J., Arai, T. and Asama, H. (2002). Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. *IEEE Transactions on Robotics and Automation*, **18**, 769–780.
- Ni, J.J. and Yang, S.X. (2011). Bioinspired neural network for real-time cooperative hunting by multirobots in unknown environments. *IEEE Transactions on Neural Networks*, **22**, 2062–2077.
- Paull, L., Saeedi, S., Seto, M. and Li, H. (2014). AUV navigation and localization: a review. *IEEE Journal of Oceanic Engineering*, **39**, 131–149.
- Polycarpou, M.M., Yang, Y. and Passino, K.M. (2001). Cooperative control of distributed multi-agent systems. *IEEE Control Systems Magazine*, **21**, 1–27.
- Reeve, R. and Hallam, J. (2005). An analysis of neural models for walking control. *IEEE Transactions on Neural Networks*, **16**, 733–742.
- Woolsey, C. and Techy, L. (2009). Cross-track control of a slender, underactuated AUV using potential shaping. *Ocean Engineering*, **36**, 82–91.
- Yang, S.X. and Meng, M. (2003). Real-time collision-free motion planning of a mobile robot using a neural dynamics-based approach. *IEEE Transactions on Neural Networks*, **14**, 1541–1552.

- Yoon, S. and Qiao, C. (2011). Cooperative search and survey using autonomous underwater vehicles (AUVs). *IEEE Transactions on Parallel and Distributed Systems*, **22**, 364–379.
- Zhu, A. and Yang, S.X. (2006). A neural network approach to dynamic task assignment of multi-robot. *IEEE Transactions on Neural Networks*, **17**, 1278–1287.
- Zhu, D.Q., Huan, 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.