THE JOURNAL OF NAVIGATION (2020), **73**, 1129–1145. © The Royal Institute of Navigation 2020
doi:10.1017/S037346332000020X

A Novel Classified Self-Organising Map Applied to Task Assignment

Yun Qu and Daqi Zhu

(*Shanghai Engineering Research Center of Intelligent Maritime Search & Rescue and Underwater Vehicles, Shanghai*)
(E-mail: zdq367@aliyun.com)

With the development of sensor technology, sensor nodes are increasingly being used in underwater environments. The strategy presented in this paper is designed to solve the problem of using a limited number of autonomous underwater vehicles (AUVs) to complete tasks such as data collection from sensor nodes when the number of AUVs is less than the number of target sensors. A novel classified self-organising map algorithm is proposed to solve the problem. First, according to the K-means algorithm, targets are classified into groups that are determined by the number of AUVs. Second, according to the self-organising map algorithm, AUVs are matched with groups. Third, each AUV is provided with the accessible order of the targets in the group. The novel classified self-organising map algorithm can be used not only to reduce the total energy consumption in a multi-AUV system, but also to give the most efficient accessible order of targets for AUVs. Results of simulations conducted to prove the applicability of the algorithm are given.

KEYWORDS

1. Autonomous Underwater Vehicles (AUVs). 2. Classified Self-Organising Map.
3. Matched. 4. Order.

Submitted: 4 July 2019. Accepted: 29 March 2020. First published online: 27 May 2020.

1. INTRODUCTION. Due to their flexibility and autonomy, AUVs (autonomous underwater vehicles) are used to explore underwater environments (Burlutskiy et al., 2012; Allotta et al., 2016; Cao et al., 2016; Yu et al., 2018; Zhang et al., 2018; Gan et al., 2019) and for scientific investigation. As a development of information technology, sensors are used to monitor conditions of the oceans. Due to limited communication distance, the data stored in these sensors are mostly collected by tools such as multi-AUV systems. Multi-AUV systems are attracting much attention in this field. Task allocation within the multi-AUV system has become a hot topic in recent years (Chang et al., 2004; Li et al., 2014; Zhu et al., 2019).

The following methods are widely used in task allocation. Conventional task allocation strategies include market mechanisms (Akkiraju et al., 2001; Sariel et al., 2006; Sotzing and Lane, 2010; Wang and Feng, 2011), contract net algorithm and distributed world model. In a market-based system, the multi-robot system is considered as an economy and each robot is considered as an agent. The algorithm has been applied to task assignment for AUVs, but the algorithm cannot guarantee the optimisation of task assignment when the destination quantity is unknown. The contract net algorithm has been used for task allocation in a multi-AUV system, and the method was used to solve the problem in an heterogeneous multi-AUV system (Li and Zhang, 2017). The distributed world model was presented to solve the problem of a cooperative robotic team under communication constraints (Tsiogkas et al., 2014). However, these methods cannot solve the problem of task allocation when the number of targets is unknown. Due to the similarity between the self-organising map (SOM) algorithm and task allocation, the SOM has been applied to task allocation (Tambouratzis, 2006; Muslim et al., 2007; Huang et al., 2012; Wang et al., 2015; Liu and Zhu, 2017; Zhu et al., 2017). The SOM neural network was first proposed by Kohonen in the 1980s (Kohonen et al., 2000). The method is used for dynamic classification of input vectors. Targets are allocated to the corresponding robot according to the Euclidean distance. The SOM algorithm (Zhu et al., 2012) effectively solves the problem of an unknown number of targets. With the development of sensor technology, a strategy is needed to solve the problems that arise when using a limited number of AUVs to complete tasks, such as data collection from sensors.

The traditional SOM algorithm has two limitations in task allocation and path planning. First, the method relies on the order of the input vector. Because the number of AUVs is less than the number of targets, there exists a problem with accessible order of the targets for the AUV. When the order of the input vector is changed, the method may cause the large energy consumption mainly because AUV has reentrant path. Second, the performance of the total task assignment is bad. Because a single AUV carries limited energy, each AUV may not be assigned too many tasks, although the AUV is the nearest to the target. At the same time, the target is assigned to the farther AUV. Hence, the traditional algorithm may cause greater energy consumption overall. The traditional SOM algorithm cannot give the accessible order of targets for AUVs in a case where the number of targets is much greater than that of AUVs.

The classified SOM algorithm is presented to solve this problem. A single AUV is matched with a corresponding group of targets, and the classified SOM method then calculates the navigation distance between the AUV and each target in the group. The target with the minimum navigation distance is the first accessible target of the AUV. The location of the AUV is then updated to the location of that target until all the targets in the group have been accessed by the AUV. This method can give a more efficient accessible order of targets for AUVs and it enables the AUVs to visit targets while guaranteeing the least total energy consumption. AUVs may have different kinematics and dynamics from unmanned aerial vehicles (UAVs), but the task allocation problems for them share certain similarities. The improved algorithm may be used in more fields where the task allocation problems share certain similarities, such as UAVs (Sujit and Beard, 2007; Xiang et al., 2017).

The structure of the paper is as follows. A description of the task allocation problem is presented in Section 2. In Section 3, the classified SOM algorithm is proposed to solve the issue. In order to prove the effectiveness of the proposed method, the simulation and its results are given in Section 4. In Section 5, the conclusion is presented.

2. DESCRIPTION OF TASK ALLOCATION PROBLEM.

In task allocation, tasks (target sensors) are effectively allocated to agents (AUVs) through a certain strategy. The

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

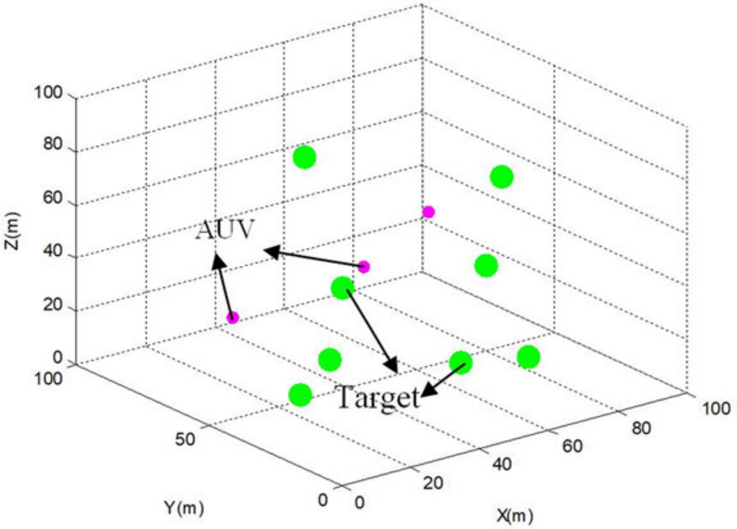

Figure 1. AUVs and targets randomly distributed.

task allocation problem is thus an assorted optimisation issue. Path planning is how to select the reasonable path and accessible order (in this case) for an AUV to access target sensors. The task allocation problem is illustrated in Figure 1, with eight targets (green dots) and three AUVs (pink dots) in the workspace. We evaluate the AUV consumption by the navigated distance from its starting position to its destination. The main issue of the multi-AUV system is how to divide the whole task into several subtasks so that AUVs can move to their designated targets along optimized paths.

3. ALGORITHM ANALYSIS.

3.1. *Traditional SOM task allocation algorithm.* SOM is a clustering and unsupervised learning algorithm for high dimensional visualisation. It is very similar to the task assignment problem, so the SOM algorithm can be used to solve the issue of task assignment. The structure of SOM is shown in Figure 2. The SOM neural network is a network consisting of two layers. The input layer is the location coordinate of the target. The output layer is the location coordinate of the AUV, and each neuron in the output layer connects through the weight vector to neurons in the input layer. According to the SOM algorithm, the task allocation problem is divided into several sub-problems including the selection of the winning neuron, the definition of the neighbourhood function and the update of the weight vector.

First, the weight vector is used to select the winning neuron, and the neighbour neurons are decided by the neighbourhood function. Then, the winner and neighbour neurons move to the target. When (in this case) the target has been accessed by an AUV, that AUV becomes free for re-assignment. When all the targets have been accessed, the task allocation and path planning are completed.

3.1.1. *Winner.* The selection of the winner is defined as follows (Zhu et al., 2012):

$$[P_j] \Leftarrow \min\{D_{kjl}, k = 1, 2, \ldots, K; j = 1, 2, \ldots J; l = 1, 2, \ldots, L\} \tag{1}$$

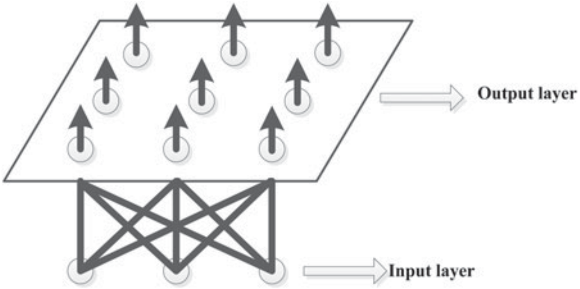

Figure 2. Structure of the SOM algorithm.

where $\left[P_j\right]$ represents that the jth neuron from the kth group is the selected winner to the lth input node. D_{kjl} is a value related to the Euclidean distance between the AUV and the target. The selection of the winner depends on the definition of the D_{kjl}. The Euclidean distance is expressed as:

$$\left|T_l - R_{jk}\right| = \sqrt{\left(x_l - w_{jkx}\right)^2 + \left(y_l - w_{jky}\right)^2 + \left(z_l - w_{jkz}\right)^2} \tag{2}$$

$T_l = (x_l, y_l, z_l)$, which is the lth target's coordinate, is the location of the lth target in the three-dimensional workspace. $R_{jk} = \left(w_{jkx}, w_{jky}, w_{jkz}\right)$, which is the coordinate of the AUV in real time, is the location of the jth neuron from the kth group. Due to the limited energy available, the parameter P is used to adjust the energy consumption of the AUV. P is defined as:

$$W = \frac{P_j - \bar{W}}{S + \bar{W}} \tag{3}$$

S is the safe distance, where the navigational energy consumption of an AUV is less than the amount of energy it carries. $\bar{W}$ is the average moving length of a team of AUVs in a certain task. Pj is the actual navigation length of the AUV. D_{kjl} is defined as:

$$D_{kjl} = \begin{cases} \left|T_l - R_{jk}\right| & 0 \le P_j \le S \\ \left|T_l - R_{jk}\right| \ (1 + W) & \mathrm{S} \le P_j \le S_{\max} \\ \infty & P_j \ge S_{\max} \end{cases} \tag{4}$$

Here, the maximal length of a single AUV navigational is represented by $S_{\max}$. There are three different cases in Equation (4). First, $0 \le P_j \le S$, where the actually navigation distance is less than the safe distance, does not need $\overline{W}$ to adjust the energy consumption. Second, $\overline{W}$ is used to adjust the energy consumption in a case where the actual navigation distance is greater than the safe distance. Finally, when the actual navigation distance is greater than the maximum navigation length, an AUV cannot be assigned to the target.

3.1.2. *Neighbourhood function.* The weights of winner and neighbour neurons are dependent on the neighbourhood function. The neighbour neurons are calculated as follows:

$$f\left(d_m, g\right) = \begin{cases} e^{-d_m^2/g^2(t)}, & d_m < \lambda \\ 0, & \text{others} \end{cases} \tag{5}$$

Radius λ represents the neighbour scope where the centre is the winning neuron. $d_m = \left|N_m - N_j\right|$ denotes the distance between the m_{th} AUV and the winning neuron. The function

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

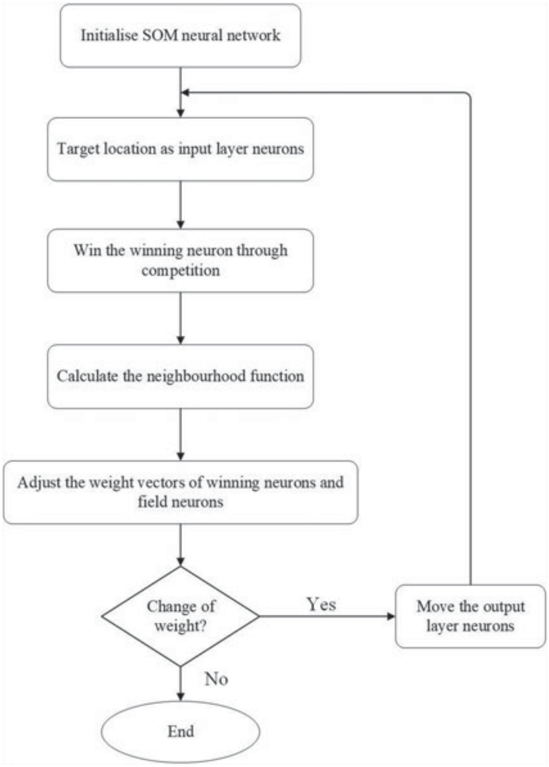
Initialise SOM neural network
Target location as input layer neurons
Win the winning neuron through competition
Calculate the neighbourhood function
Adjust the weight vectors of winning neurons and field neurons
Change of weight?
Yes
Move the output layer neurons
No
End

Figure 3. Flow chart of the SOM algorithm.

$g^2(t)$ is defined as follows:

$$g^2(t) = (1 - \partial)^t g_o \tag{6}$$

∂ is a fixed learning rate and $\partial < 1$. The calculation time is decreased as the ∂ increases. The update rule is defined as:

$$R_{jk}(t+1) = \begin{cases} T_l, & D_{kjl} \leq D_{\min} \\ R_{jk}(t) + \partial \times f(d_m, g) \times (T_l - R_{jk}(t)), & D_{kjl} > D_{\min} \end{cases} \tag{7}$$

In one case, where $D_{\min}$, which represents the termination condition of iterations, is greater than D_{kjl}, the weight is replaced by the target coordinate. In another case, the weight's adjustment depends on the learning speed ∂, neighbourhood function $f(d_m, g)$ and the location of the winner and neighbours.

A flow chart of the SOM algorithm is presented in Figure 3. First, it initialises the location coordinates of the AUV and targets. Second, the location coordinates of the targets are input into the input layer and the location coordinates of the AUV are input into the output layer. According to the Euclidean distance, the winning neuron is selected through competition. Then the method calculates the neighbourhood function and adjusts the weight vectors. If the weight vectors change, the AUV moves towards the target point. When the weight vectors do not change, the task assignment and path planning are complete.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

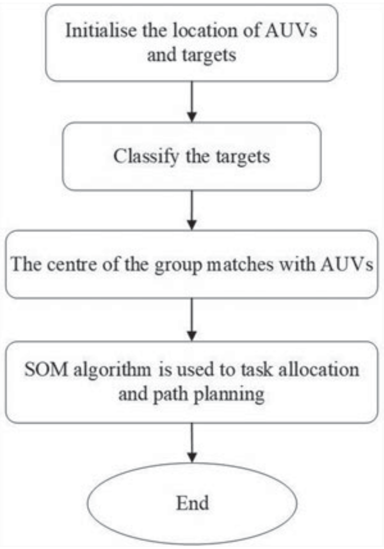
Initialise the location of AUVs and targets
Classify the targets
The centre of the group matches with AUVs
SOM algorithm is used to task allocation and path planning
End

Figure 4. Flow chart of the CSOM algorithm.

The traditional SOM method has two limitations in task allocation and path planning, as explained above, in its inability to solve the problem of accessible order of targets for the AUV and to take energy consumption into account.

3.2. *Classified SOM algorithm.* In order to solve the problem of accessible order of targets for the AUVs and to minimise the total energy consumption of the AUV system, the classified SOM (CSOM) algorithm is proposed. An algorithm flow chart of the CSOM method is presented in Figure 4. First, the location coordinates of AUVs and targets are initialised. According to the K-means algorithm, targets are classified into groups which are determined by the number of AUVs. The centres of groups are then matched with AUVs. The SOM algorithm is then used for task allocation and path planning.

3.2.1. *Classify the targets.* The K-means algorithm is used to classify the targets. The number of groups depends on the number of AUVs. The flow chart of this method is shown in Figure 5. First, the location coordinates of centres and targets are initialised. D_{kn}, which denotes the distance between centres and targets, is defined as:

$$D_{kn} = \sqrt{(C_k - T_n)^2} \tag{8}$$

C_k represents the kth centre and the number of centres depends on the number of AUVs, as shown in Figure 6. T_n represents the coordinate location of the nth target. S_{kn} is defined as:

$$S_{kn} \Leftarrow \min\{D_{kn}, k = 1, 2, \ldots, K; n = 1, 2, \ldots, N\} \tag{9}$$

S_{kn} belongs to the kth group, because the nth target is the nearest to the kth centre. Then, according to the location of targets in the group, the method calculates the centre of each group. When the centres are fixed, the grouping is completed. C_k represents the kth centre of group. An example is shown in Figure 6, where pink dots represent five AUVs and green dots represent 20 targets. In this example, based on the number of AUVs, the targets are classified into five groups.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

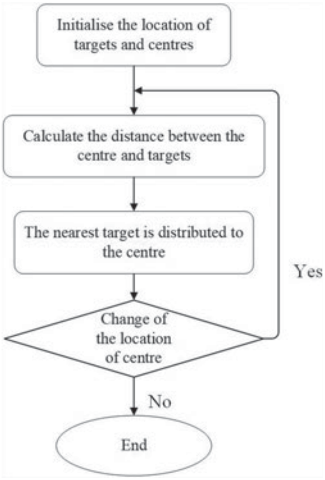
Initialise the location of targets and centres
Calculate the distance between the centre and targets
The nearest target is distributed to the centre
Change of the location of centre
Yes
No
End

Figure 5. Flow chart of K-means.

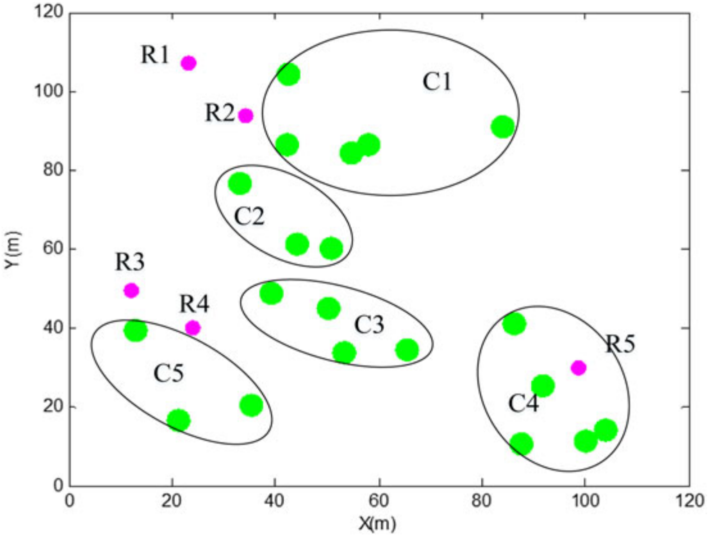

Figure 6. Classification of targets.

3.2.2. *AUVs matched with groups.* When the targets have been classified into several groups, the centres of the groups are matched with AUVs. M_{ki} is defined as:

$$M_{ki} = \sqrt{(C_k - R_i)^2} \tag{10}$$

where M_{ki} represents the Euclidean distance between the kth centre and the$i$th robot (AUV). CR_{ki} represents the ith robot which is the nearest to the kth centre and is defined as:

$$CR_{ki} \Leftarrow \min\{M_{ki}, k = 1, 2, \ldots, K\} \tag{11}$$

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

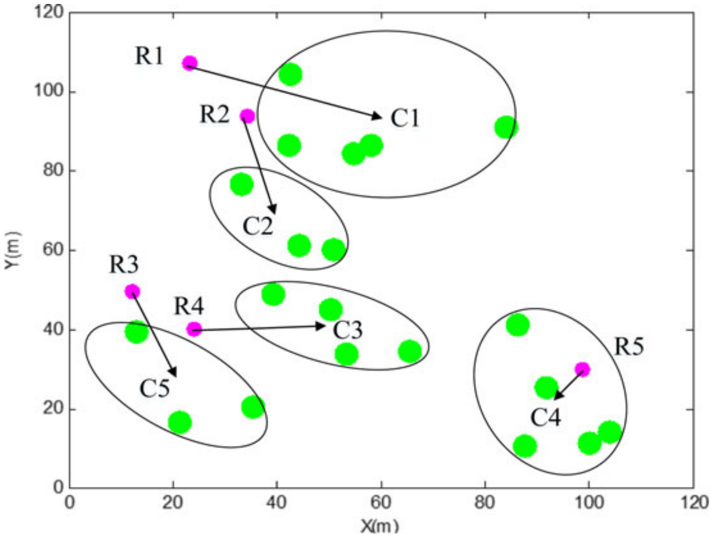

Figure 7. AUVs matched with the groups of targets.

An example is shown in Figure 7, where again the pink dots represent five AUVs and green dots represent 20 targets. C_k represents the kth centre of group. According to the Euclidean distance between AUVs and centres, AUVs are matched with the centres of the several groups.

3.2.3. *Improved SOM method in task allocation.* After the AUVs have been matched with groups, the improved SOM method is used to allocate tasks to the AUV in each group. The improved SOM method changes the connected relation between the AUV and targets. The structure of the improved SOM method is presented in Figure 8. The input layer neuron is dynamic. A single AUV connects with all targets. The AUV is allocated to the targets in the group according to the shortest actual navigation distance to each. The target that represents the minimum navigation distance in the group is the first accessible target of the AUV. When an AUV accesses a target, the location of the AUV is updated to the location of that target until all targets in the group have been accessed by the AUV. As shown in Figure 9, the pink dot represents the AUV and the green dot represents the target. α represents the angle between the x-axis and the black dotted line that connects the AUV and the target point at the initial time. θ is the initial orientation of the AUV and r is the turning radius of the AUV. β is defined as follows:

$$\beta = \theta - \alpha \tag{12}$$

The AUV has two turning paths, shown as the red solid line and the red dotted line in Figure 9. The AUV is steered in the direction of $|\beta|$ at decreasing distance until $\beta = 0$. The actual path travelled by the AUV is the red solid line plus black line in Figure 9. D_{uj} is defined as:

$$D_{uj} = \Delta U + \left| R_u - T_j \right| \tag{13}$$

where D_{uj} represents the actual navigation distance. ΔU is the turning distance of the uth AUV when turning to $\beta = 0$. $\left| R_u - T_j \right|$ is the Euclidean distance between the$j$ th target and

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

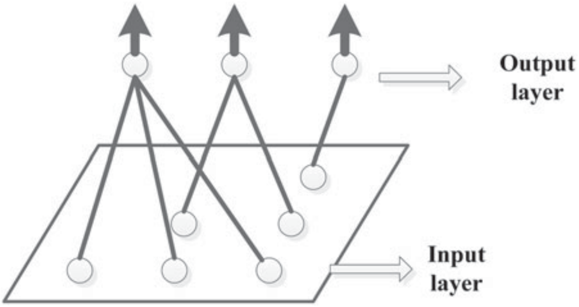

Figure 8. Structure of the improved SOM algorithm.

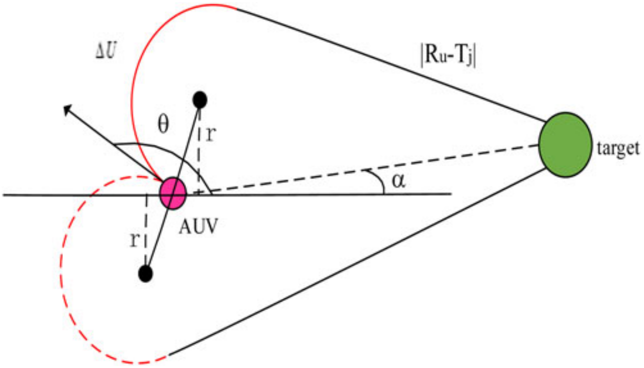

Figure 9. Actual navigation distance of AUV.

the *u*th turning AUV. The accessible order of targets to the AUV depends on the [*Ai*]. The selection of the nearest target is defined as

$$[Ai] \Leftarrow \min\{D_{uj}, j = 1, 2, \ldots J\} \tag{14}$$

where [*Ai*] represents the nearest target to the *u*th AUV. *J* is the number of targets in the corresponding group. The method then calculates the neighbourhood function and adjusts the weight vectors. If the weight vectors are changed, the AUV moves towards the target point. When the location of the AUV is updated to the location of the target, the AUV becomes free and the remaining targets are input into the input layer. When all targets have been accessed by the AUV, the task allocation and path planning are completed for that group.

4. SIMULATION.

Due to random errors in the position, speed, angle and turning radius of AUVs, the error datum are processed with extended Kalman filter (Loebis et al., 2004; Wu et al., 2019). For example, in Figure 10, the black line represents the actual value of the AUV, the blue line represents the measured value and the red line represents the extended Kalman filter estimated value. To increase the accuracy of the error datum, all datum are processed with the extended Kalman filter.

4.1. *Two-dimensional plane simulation results.*

The study of the three-dimensional algorithm is based on the study of the two-dimensional algorithm. The simulation results of the two-dimensional algorithm reflect the effectiveness more intuitively. The location

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

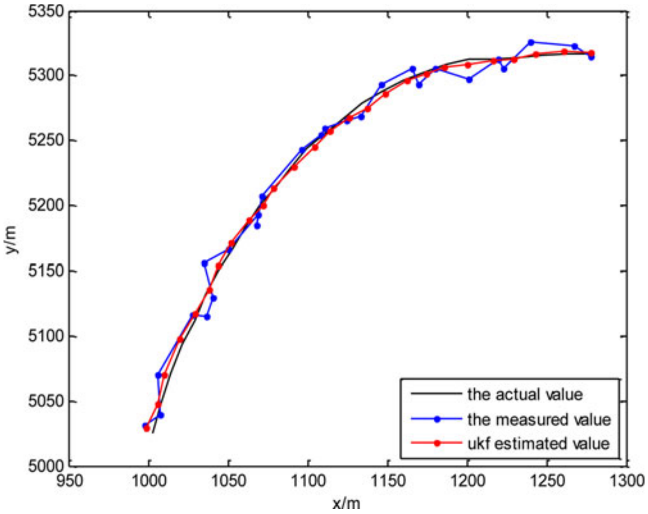

Figure 10. Extended Kalman filter algorithm.

coordinates of AUVs are set to random number seed 5. The location coordinates of targets are set to random number seed 15. The turning radius of the AUV is 0·8 and its running speed is 1·5 m/s. For R1, the angle between R1 and the x-axis is θ_1 and θ_1 is 150°. The angles of R2 and so on are shown in Figures 11 and 12. The traditional SOM is presented in Figure 11. There are 25 targets (green dots) and five AUVs (R1 ... R5) (pink dots) in the workspace. Due to the problem of the input order in the traditional method, when the target is input in the input layer, the winner (AUV), which is the nearest to the target, is selected for the target. Although the target is allocated to the nearest AUV, another target, which is not allocated, is nearer to the AUV. This can cause more energy consumption for the AUV. For example, R1 is nearest to T1, but R1 accesses T3 first. The accessible order of targets for R1 clearly causes more energy consumption. The total distance travelled by the AUVs (as a proxy for energy consumption) is 725·6460 m in the traditional method.

In Figure 12, the location coordinates of the five AUVs (pink dots) and targets are the same as in Figure 11. Based on the number of AUVs, the targets are now divided into five groups. The green dots are allocated to R1, the blue dots are allocated to the R2, the black dots are allocated to the R3, the yellow dots are allocated to the R4, and the red dots are allocated to the R5. For example, T11, T13 and T14 are allocated to R2. According to the actual navigation distance between the AUV and the target, the CSOM method can give the most efficient accessible order of targets for the AUV. The total distance travelled by the AUVs is 599·9616 m in the CSOM method. The accessible order of targets to the AUVs is more efficient in the CSOM than the traditional SOM, and the CSOM method can also save energy consumption by reducing the distance travelled by the AUVs.

Figures 13 and 14 illustrate the effects on energy consumption in terms of distance travelled. In Figure 13 the x-axis represents the number of targets and the y-axis represents the total energy consumption of the AUVs. The red line represents the total energy consumption of the CSOM method and the black line represents that of the SOM method. There is an intersection of the two lines and the abscissa of the intersection is seven. When the number of targets is more than seven, the total energy consumption in the CSOM is less than in the traditional SOM. When the number of targets is less than seven, the total

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

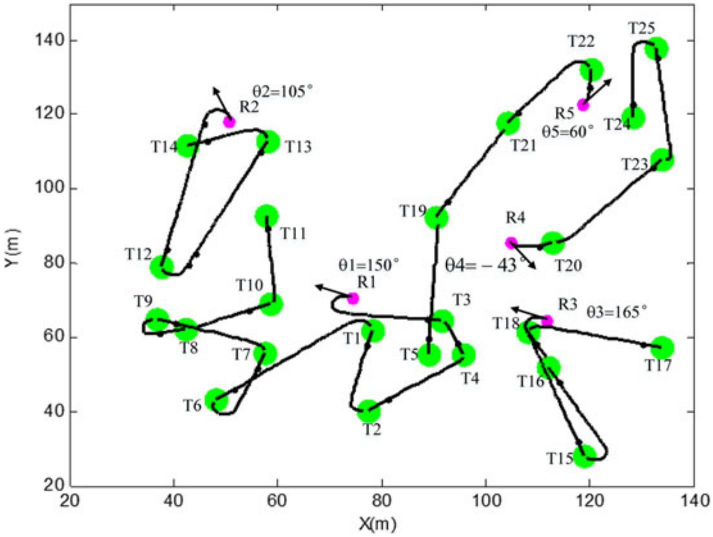

Figure 11. The traditional SOM method.

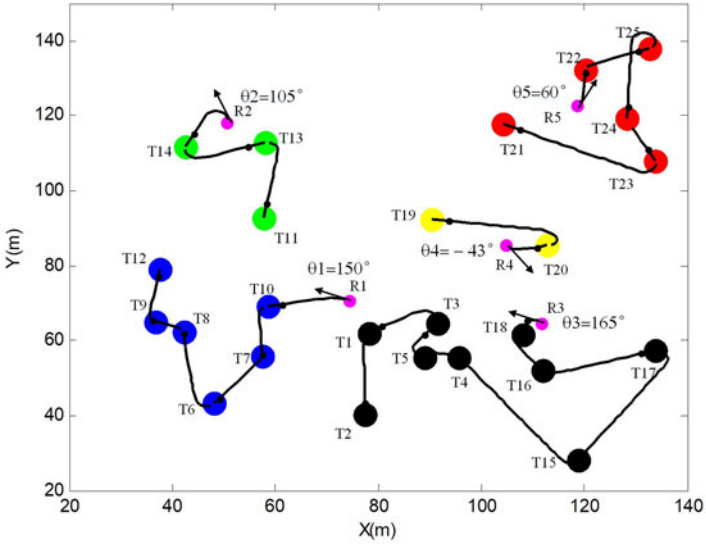

Figure 12. The CSOM method.

energy consumption in the traditional SOM is less than in the CSOM. Because there are three AUVs in the workspace, there is no need to classify targets because the number of AUVs is close to the number of targets. As the number of targets increases, the energy consumption gap between the CSOM method and the traditional SOM method increases. In Figure 14, the x-axis represents the number of AUVs and the y-axis represents the total energy consumption. The red line represents the total energy consumption of the CSOM method and the black line represents that of the traditional SOM method. There are 40 targets and some AUVs in the workspace environment. In Figure 14, as the number of AUVs increases, the total energy consumption in the CSOM is less than the SOM.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

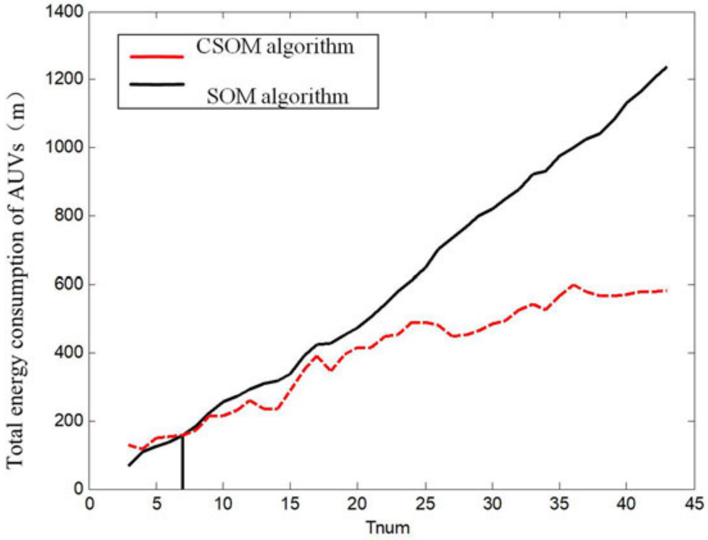

Figure 13. Total energy consumption of AUVs by number of targets (*Tnum* = 0–45).

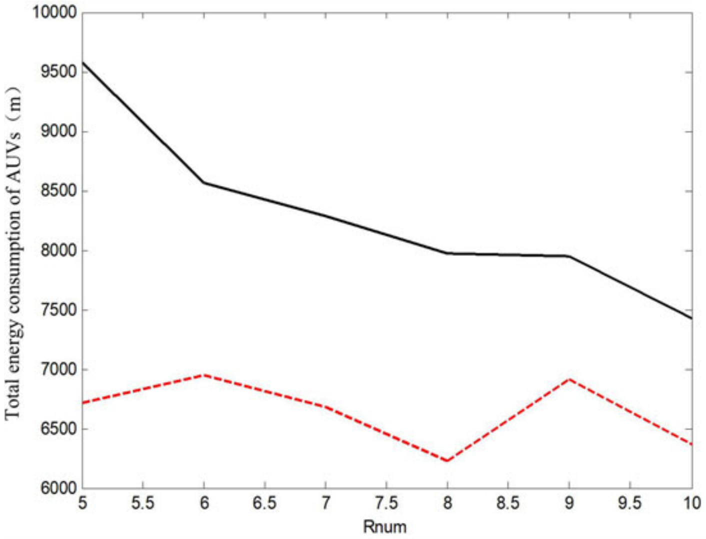

Figure 14. Total energy consumption of AUVs by number of AUVs (*Rnum* = 5–10).

Table 1. Initial orientation of the robot (AUV) in three-dimensional plane.

Robot	Angle between robot and *x*-axis	Angle between robot and *z*-axis
R1	−20°	15°
R2	−60°	10°
R3	−120°	−20°

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

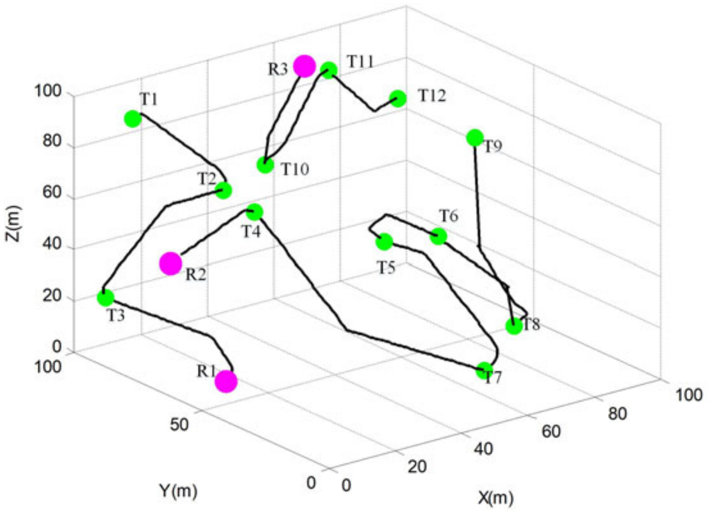

Figure 15. The traditional SOM method in three-dimensional plane.

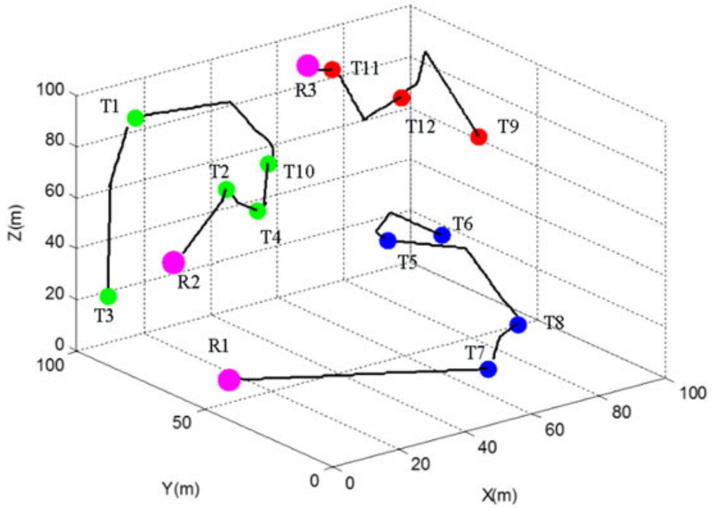

Figure 16. The CSOM method in three-dimensional plane.

The CSOM method can save energy in the case where the number of targets is much greater than the number of AUVs, and it can also give a more efficient accessible order of targets than in the traditional SOM method.

4.2. *Three-dimensional plane simulation results.* The location coordinates of AUVs are set to random number seed 8. The location coordinates of targets are set to random number seed 4. The turning radius of the AUV is 0·8 and its running speed is 1·5. The initial angles of the AUVs (R1–R3) are illustrated in Table 1. The traditional SOM is presented in Figure 15. There are 12 targets (green dots) and three AUVs (pink dots) in the workspace. With SOM, the same problem exists in the three-dimensional plane as in

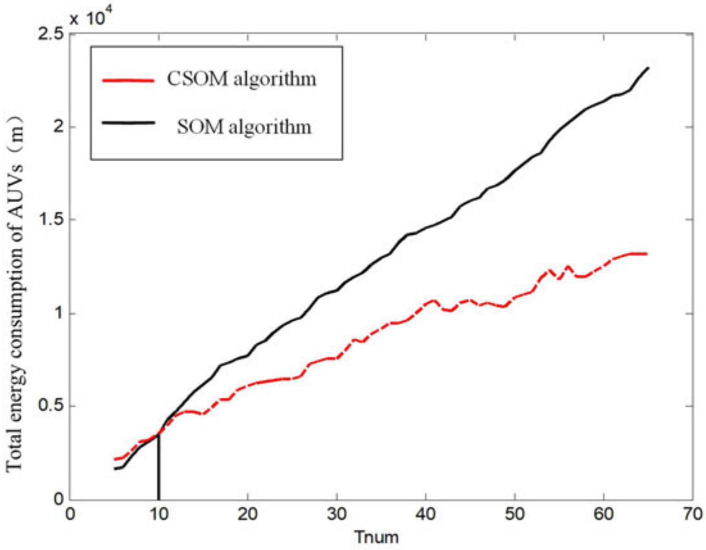

Figure 17. Total energy consumption of AUVs by number of targets (*Tnum* = 0–70).

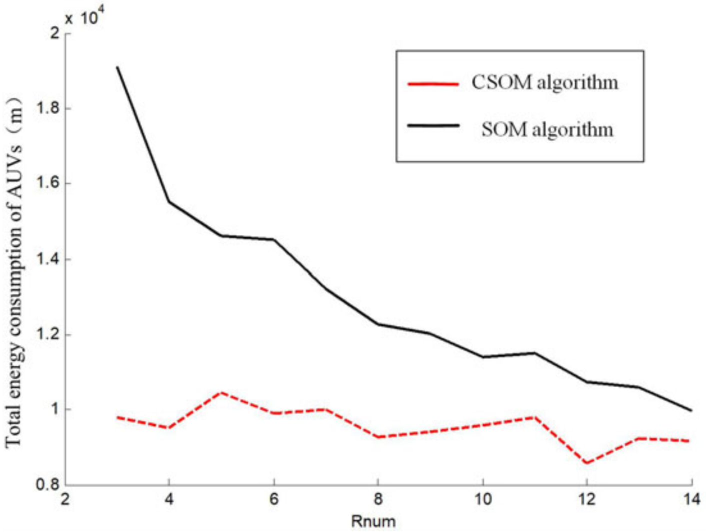

Figure 18. Total energy consumption of AUVs by number of AUVs (*Rnum* = 2–14).

the two-dimensional plane. For example, R3 is nearest to T11, but it accesses T10 first, which clearly causes more energy consumption. The total distance travelled by the AUVs is 631·0219 m in the traditional SOM method. In Figure 16, the location coordinates of AUVs and targets are the same as in Figure 15. The pink dots represent AUVs and the other coloured dots represent targets. Based on the number of AUVs, the targets are now divided into three groups. The blue target dots are allocated to R1, the green target dots are allocated to R2, and the red target dots are allocated to R3. For example, T11, T12 and T9 are allocated to R3. The total distance travelled by the AUVs is 440.3366 m in the CSOM method.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

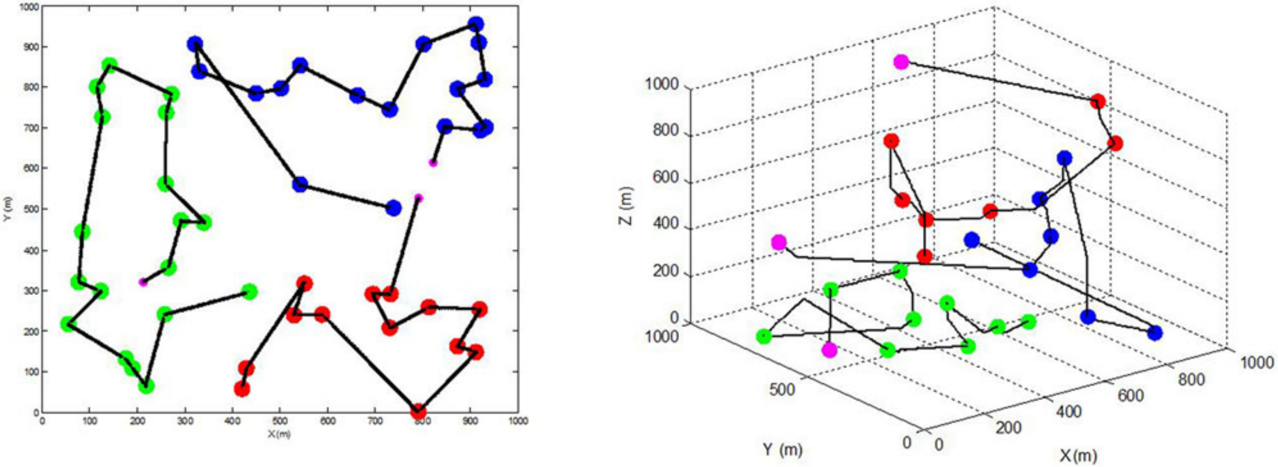

Figure 19. CSOM method where number of targets greatly exceeds number of AUVs: two-dimensional plane (left) and three-dimensional plane (right).

4.3. *Large number simulation results.* Further simulations were conducted with larger numbers of AUVs and targets to justify the methodology proposed. Figures 17 and 18 illustrate the energy consumption. Figure 19 models the CSOM method where the number of targets (40) greatly exceeds number of AUVs (5) in two and three dimensions. In Figure 17 the x-axis represents the number of targets (0–70) and the y-axis represents the total energy consumption of the AUVs in terms of distance travelled. The red line represents the total energy consumption of the CSOM method and the black line represents that of the SOM method. There is an intersection of the two lines and the abscissa of the intersection is 10. When the number of targets is more than 10, the total energy consumption in the CSOM is less than that in the SOM. When the number of targets is less than 10, the total energy consumption in the SOM is less than that in the CSOM. If there were five AUVs and 10 targets in the workspace, it would not be necessary to classify the targets because the number of AUVs would be close to the number of targets. As the number of targets increases, the energy consumption gap between the CSOM method and the SOM method increases. In Figure 18, the x-axis represents the number of AUVs (2–14) and the y-axis represents the total energy consumption in terms of distance travelled. The red line represents the total energy consumption of the CSOM method and the black line represents the total energy consumption of the SOM method. There are 40 targets and some AUVs in the workspace environment. In Figure 18, as the number of AUVs increases, the total energy consumption in the CSOM becomes less than that in the SOM.

The concrete analyses are the same as in Figures 12 and 16. As the gap between the number of AUVs and the number of targets increases, the different levels of energy consumption are illustrated in Figures 13, 14, 17 and 18.

5. CONCLUSION.

In this paper, a classified SOM algorithm is proposed. The CSOM algorithm not only gives a more efficient accessible order of targets for the AUVs than the SOM method, but it can also reduce energy consumption. Given that each AUV carries a limited supply of energy for each mission, the CSOM method can enable AUVs to access more targets than the SOM method in the same environment. Future research will explore the effect of ocean currents on the task assignment problem for AUV systems. An ocean current is described by its velocity and direction in the actual underwater environment. By measuring the ocean current in a certain period of time, the ocean current average is obtained. The ocean current average is used as the constant ocean current for modelling the underwater working environment. The CSOM method can thus be extended to the ocean environment. It can also be extended to cases where targets have different priorities.

ACKNOWLEDGEMENTS

This project is supported by the National Natural Science Foundation of China (U1706224, 51575336, 91748117) and the Creative Activity Plan for Science and Technology Commission of Shanghai (18DZ2253100, 18JC1413000, 18DZ1206305)

REFERENCES

Akkiraju, R., Keskinocak, P., Murthy, S. and Wu, F. (2001). An agent-based approach for scheduling multiple machines. *Applied Intelligence*, 14(2), 135–144.

Allotta, B., Caiti, A., Costanzi, R., Fanelli, F., Fenucci, D., Meli, E. and Ridolfi, A. (2016). A new AUV navigation system exploiting unscented Kalman filter. *Ocean Engineering*, 113, 121–132.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press

Burlutskiy, N., Touahmi, Y. and Lee, B. L. (2012). Power efficient formation configuration for centralized leader-follower AUVs control. *Journal of Marine Science and Technology*, 17(3), 315–329.

Cao, X., Zhu, D. and Yang, S. X. (2016). Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments. *IEEE Transactions on Neural Networks and Learning Systems*, 27(11), 2364–2374.

Chang, Z., Bian, X., Yan, Z. and Wang, Y. (2004). Study of AUV's real-time task coordination method. *Journal of Harbin Engineering University*, 25(4), 472–476.

Gan, W., Zhu, D., Hu, Z., Yang, L., Shi, X. and Chen, Y. (2019). Model predictive adaptive constraint tracking control for underwater vehicles. *IEEE Transactions on Industrial Electronics*, doi:10.1109/TIE.2019.2941132

Huang, H., Zhu, D. and Yuan, F. (2012). Dynamic Task Assignment and Path Planning for Multi-AUV System in 2D Variable Ocean Current Environment. *Chinese Control and Decision Conference(CCDC)*, Taiyuan, China.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V. and Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3), 574–585.

Li, J. and Zhang, K. (2017). Heterogeneous multi-AUV cooperative task allocation based on improved contract net algorithm. *Journal of Unmanned Undersea Systems*, 25(5), 418–423.

Li, J., Zhang, R. and Yang, Y. (2014) Research on Route Obstacle Avoidance Task Planning Based on Differential Evolution Algorithm for AUV. *International Conference in Swarm Intelligence(ICSI)*, Hefei, China.

Liu, Y. and Zhu, D. (2017). Task assignment and path planning of AUV system based on Glasius bio-inspired self-organizing map neural network algorithm. *System Simulation Technology*, 13(3), 231–234.

Loebis, D., Sutton, R., Chudley, J. and Naeem, W. (2004). Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control Engineering Practice*, 12(12), 1531–1539.

Muslim, M. A., Ishikawa, M. and Furukawa, T. (2007). Task segmentation in a mobile robot by MnSOM: a new approach to training expert modules. *Neural Computing and Applications*, 16(6), 571–580.

Sariel, S., Balch, T. and Stack, J. (2006). Distributed multi-AUV coordination in naval mine countermeasure missions. *IEEE Robotics and Automation Magazine*, 15(1), 45–52.

Sotzing, C. C. and Lane, D. M. (2010). Improving the coordination efficiency of limited communication multi-AUV operations using a multi-agent architecture. *Journal of Field Robotics*, 27(4), 412–429.

Sujit, B. and Beard, R. (2007). Distributed Sequential Auctions for Multiple UAV Task Allocation. *American Control Conference (ACC)*, New York, USA.

Tambouratzis, G. (2006). Assessing the effectiveness of feature groups in author recognition tasks with the SOM model. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews*, 36(2), 249–259.

Tsiogkas, N., Papadimitriou, G., Saigol, Z. and Lane, D. (2014). Efficient Multi-AUV Cooperation Using Semantic Knowledge Representation for Underwater Archaeology Missions. *Oceans - St. John's*, St. John's, NL, Canada.

Wang, Z. and Feng, X. (2011) A Cooperative Simulation System for AUV Based on Multi-Agent. *IEEE International Conference on Virtual Reality and Visualization(ICVRV)*, Beijing, China.

Wang, C., Chen, C. and Hung, K. N. (2015). Toward a new task assignment and path evolution (TAPE) for missile defense system (MDS) using intelligent adaptive SOM with recurrent neural networks (RNNs). *IEEE Transactions on Cybernetics*, 45(6), 1134–1145.

Wu, Y., Ta, X., Xiao, R., Wei, Y., An, D. and Li, D. (2019). Survey of underwater robot positioning navigation. *Applied Ocean Research*, doi:10.1016/j.apor.2019.06.002

Xiang, X., Liu, C., Su, H. and Zhang, Q. (2017). On decentralized adaptive full-order sliding mode control of multiple UAVs. *ISA Transactions*, 71(2), 196–205.

Yu, C., Xiang, X., Lapierre, L. and Zhang, Q. (2018). Robust magnetic tracking of subsea cable by AUV in the presence of sensor noise and ocean currents. *IEEE Journal of Oceanic Engineering*, 43(2), 311–322.

Zhang, Q., Zhang, J., Chemori, A. and Xiang, X. (2018). Virtual submerged floating operational system for robotic manipulation. *Complexity*, doi:10.1155/2018/9528313

Zhu, D., Huang, H. and Yang, S. X. (2012). 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(2), 504–514.

Zhu, D., Liu, Y. and Sun, B. (2017). Task assignment and path planning of a multi-AUV system based on a Glasius bio-inspired self-organising map algorithm. *Journal of Navigation*, 71(2), 1–15.

Zhu, D., Qu, Y. and Yang, S. X. (2019). Multi-AUV SOM task allocation algorithm considering initial orientation and ocean current environment. *Frontiers of Information Technology and Electronic Engineering*, 20(3), 330–341.

https://doi.org/10.1017/S037346332000020X Published online by Cambridge University Press