# Letter

## Model-Free Formation Control of Autonomous Underwater Vehicles: A Broad Learning-Based Solution

Wenqiang Cao, Jing Yan, *Senior Member, IEEE*, Xian Yang, Xiaoyuan Luo, and Xinping Guan, *Fellow, IEEE*

Dear Editor,

We develop a broad learning-based algorithm to enforce the formation control of AUVs. Compared with the deep learning (DL) based formation solutions, our solution employs the broad learning system (BLS) to remodel the learning framework without a retraining process, such that the learning structure can be simplified and the training time can be reduced. Finally, simulation and experimental studies are both performed to verify the effectiveness.

**Related work:** Recently, formation control of multiple AUVs has received increasing attention in marine applications (see [1]−[3] and the references therein). Many DL-based formation controllers have been developed to relax the dependence of accurate model parameters, e.g., [4] and [5]. Nonetheless, the DL-based formation controllers suffer from a time-consuming training process due to a large number of parameters in filters and layers, which limits their further implementation in resource-constrained AUV. With this problem in hand, we notice that a new concept of discriminative learning framework termed as BLS [6] can provide us with an effective solution. Along with this, a BLS-based AUV tracking controller was developed in [7], through which the ridge regression can be employed to seek the local optimization solution. Followed by this, [8] combined the adaptive dynamic programming (ADP) into the framework of BLS, such that a global optimization tracking solution can be obtained by AUV. However, the tracking solution in [8] can not remodel the learning framework with a less cost on account of the weight iteration scheme. Per knowledge of the authors, how to employ the BLS to design a model-free formation algorithm with remouldable network structure is still an open issue for AUVs.

In this work, we aim to present a BLS-based algorithm to enforce the model-free formation control of AUVs. We first employ the inputs of AUVs to construct a set of mapped features, and hence, the enhancement nodes are constructed by connecting the mapped features in different groups. To the end, an incremental learning-based formation algorithm with ADP and BLS is designed for AUVs major contributions of this letter lies in two aspects: 1) Construct a BLS-based formation framework, which can simplify the learning structure and save the training time; 2) Employ the ADP and BLS to design a model-free formation algorithm, such that the remouldable network structure and the global optimization can be guaranteed.

Corresponding author: Jing Yan.

Citation: W. Q. Cao, J. Yan, X. Yang, X. Y. Luo, and X. P. Guan, "Model-free formation control of autonomous underwater vehicles: A broad learning-based solution," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 5, pp. 1325–1328, May 2023.

W. Q. Cao, J. Yan, and X. Y. Luo are with the Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China (e-mail: cwq@stumail.ysu.edu.cn; jyan@ysu.edu.cn; xyluo@ysu.edu.cn).

X. Yang is with the Institute of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China (e-mail: xyang@ysu.edu.cn).

X. P. Guan is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xpguan@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123165

**Problem formulation:** Denote $\eta_i = [x_i, y_i, z_i, \varphi_i]^T$ and $\mathbf{v}_i = [u_i, v_i, w_i, r_i]^T$ as the position and velocity vectors of AUV $i$, respectively. Besides that, $\tau_i = [\tau_{u_i}, \tau_{v_i}, \tau_{w_i}, \tau_{r_i}]^T$ is the control input vector. The elements in these vectors describe the states in surge, sway, depth, and yaw, respectively. Based on this and noting with [9]−[11], the dynamics model of AUV $i$ is expressed as

$$\begin{aligned} &\dot{\eta}_i = \mathbf{J}_i(\varphi_i)\mathbf{v}_i \\ &\mathbf{M}_i\dot{\mathbf{v}}_i + \mathbf{C}_i(\mathbf{v}_i)\mathbf{v}_i + \mathbf{D}_i(\mathbf{v}_i)\mathbf{v}_i = \tau_i \end{aligned} \tag{1}$$

where $i \in \{1, \ldots, N\}$. Meanwhile, $\mathbf{J}_i(\varphi_i)$ denotes the rotation matrix, $\mathbf{M}_i \in \mathbb{R}^{4\times4}$ represents the inertia matrix, $\mathbf{C}_i(\mathbf{v}_i) \in \mathbb{R}^{4\times4}$ is the coriolis-centripetal matrix, and $\mathbf{D}_i(\mathbf{v}_i) \in \mathbb{R}^{4\times4}$ is the damping matrix. Define $T$ as the sampling interval and $\zeta_i(k) = [\eta_i(k); \mathbf{v}_i(k)]$, then the above model can be described by the discrete form, i.e., at the $(k+1)$-th sampling,

$$\zeta_i(k+1) = f_i(\zeta_i(k)) + h_i(\zeta_i(k))\tau_i(k) \tag{2}$$

with

$$f_i(\zeta_i) = \begin{bmatrix} \eta_i + T\mathbf{J}_i(\varphi_i)\mathbf{v}_i \\ \mathbf{v}_i - T\mathbf{M}_i^{-1}(\mathbf{C}_i(\mathbf{v}_i)\mathbf{v}_i + \mathbf{D}_i(\mathbf{v}_i)\mathbf{v}_i) \end{bmatrix} \tag{3}$$

$$h_i(\zeta_i) = \begin{bmatrix} \mathbf{0}_{4\times4} \\ T\mathbf{M}_i^{-1} \end{bmatrix}. \tag{4}$$

In this letter, the topology relationship of AUVs can be described by an undirected graph $\mathcal{G} \in (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the vertex set of AUVs and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}\}$ is the communication link set. Note that the link $(i, j)$ denotes AUV $i$ can receive the information from AUV $j$, and the reverse is feasible. The neighbor set of AUV $i$ is denoted by $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Based on this, the adjacency matrix of AUVs is defined as $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N\times N}$, where $a_{ij} = 1$ if $j \in \mathcal{N}_i$ and $a_{ij} = 0$ otherwise.

Particularly, let $\mathbf{r}_{ij}$ denotes the desired relative position vector between AUV $i$ and $j$. Meanwhile, $\zeta_r$ is the position vector of the target point. Accordingly, the formation objectives are to achieve:

1) Target tracking: $\zeta_1 \underset{k\to\infty}{\to} \zeta_r$, AUV 1 directly interacts with target.

2) Formation keeping: $\zeta_i \underset{k\to\infty}{\to} \zeta_j \underset{k\to\infty}{} + \mathbf{r}_{ij}$, AUV $j \in \{1, \ldots, N\}$.

**Design and analysis:** This section presents a BLS-based algorithm to enforce the model-free formation control of AUVs. To this end, we define the following one-step cost:

$$\begin{aligned} &g_i(\zeta_i(k), \tau_i(k)) \\ &\quad = \sum_{j\in\mathcal{N}_i} \mathbf{e}_{f,i}^T(k)\mathbf{Q}_{f,i}\mathbf{e}_{f,i}(k) + \mathbf{e}_{t,i}^T(k)\mathbf{Q}_{t,i}\mathbf{e}_{t,i}(k) + \tau_i^T(k)\mathbf{R}_i\tau_i(k) \end{aligned} \tag{5}$$

where $i \in \{1, \ldots, N\}$, $\mathbf{e}_{f,i}(k) = \zeta_i(k) - \zeta_j(k) - \mathbf{r}_{ij}$ and $\mathbf{e}_{t,i}(k) = \zeta_i(k) - \zeta_r(k)$, which denote the formation error and tracking error of AUV $i$, respectively. $\mathbf{Q}_{f,i} \in \mathbb{R}^{8\times8}$ is a positive definite matrix. If $i = 1$, $\mathbf{Q}_{t,1} \in \mathbb{R}^{8\times8}$ is a positive definite matrix, otherwise, $\mathbf{Q}_{t,i} = \mathbf{0}$. $\mathbf{R}_i = \text{diag}([r_{u_i}\ r_{v_i}\ r_{w_i}\ r_{r_i}])$ is a symmetric positive definite matrix and $r_\wp$ for $\wp \in \{u_i, v_i, w_i, r_i\}$is the element of $\mathbf{R}_i$.

Define $\Omega = \{u_i, v_i, w_i, r_i\}$. The value function of AUV $i$ is

$$J_i(\zeta_i(k)) = g(\zeta_i(k), \tau_i(k)) + J_i(\zeta_i(k+1)). \tag{6}$$

By the Bellman's optimality theory, (6) can be reorganized as

$$J_i^*(\zeta_i(k)) = \min_{\tau_i(k)} \{g_i(\zeta_i(k), \tau_i(k)) + J_i(\zeta_i(k+1))\}. \tag{7}$$

In order to solve the above problem, the optimal control policy $\tau_i^*(k)$ can be obtained through $\partial J^*(\hat{\zeta}_i(k))/\partial\tau_i^*(k) = 0$, i.e.,

$$\tau_i^*(k) = -\frac{1}{2}\mathbf{R}_i^{-1}h_i^T(\zeta_i(k))\frac{\partial J_i^*(\zeta_i(k+1))}{\partial\tau_i(k+1)}. \tag{8}$$

Due to the complex marine environment, it is impossible to obtain $h_i(\zeta_i(k))$ accurately. To solve this issue, one introduces the estimated control policy $\tau_i^s$ into (2), and then (2) can be rearranged as

$$\zeta_i(k+1) = f_i(\zeta_i(k)) + h_i(\zeta_i(k))\left(\tau_i(k) - \tau_i^s(k)\right) + h_i(\zeta_i(k))\tau_i^s(k) \tag{9}$$

where $\tau_i^s$ denotes the estimated control policy in the $s$-th iteration and $\tau_i$ is an admissible policy for the learning procedure. By the definition of derivative and noting with (8) and (9), one has

$$J_i^s(\zeta_i(k+1)) - J_i^s(\zeta_i(k)) = -g_i\left(\zeta_i(k), \tau_i^s(k)\right) - 2(\tau_i^{s+1}(k))^T \mathbf{R}_i\left(\tau_i(k) - \tau_i^s(k)\right) \tag{10}$$

where $\mathbf{A}_i(\zeta_i) = \begin{bmatrix} T\mathbf{J}_i^s(\varphi_i)\mathbf{v}_i \\ -T\mathbf{M}_i^{-1}(\mathbf{C}_i(\mathbf{v}_i)\mathbf{v}_i + \mathbf{D}_i(\mathbf{v}_i)\mathbf{v}_i) \end{bmatrix}$, $J_i^s(\zeta_i(k))$ is the value function in $s$-th iteration and $\nabla J_{\zeta_i}^s(\zeta_i(k)) = \frac{\partial J_i^s(\zeta_i(k))}{\partial \zeta_i(k)}$. Thereby, the formation problem turns to the problem of solving (10).

It can be seen from (10) that $J_i^*(\zeta_i(k+1))$ and $\tau_i^s$ are unknown. In view of this, we introduce the BLS and ADP based critic-actor networks to smoothly approximate $J_i^*(\zeta_i(k+1))$ and $\tau_i^s$, respectively. There are three steps to generate BLS-based network structure. Let $\mathbf{E}_i(k) = \mathbf{e}_{f,i}(k) + b_i\mathbf{e}_{t,i}(k)$ represents the state error of AUV $i$. Of note, $b_i = 0$ when $\mathbf{Q}_{t,i} = \mathbf{0}$, otherwise $b_i = 0$.

Firstly, $\mathbf{E}_i(k)$ is used to obtain the "feature nodes" $\mathbf{Z}_\hbar(k)$, through which the "enhancement nodes" $\mathbf{H}_t(k)$ can be acquired by stochastically weighting and activating. Note that $\mathbf{Z}_\hbar(k)$ and $\mathbf{H}_t(k)$ are used as the activation functions for BLS architecture, i.e.,

$$\mathbf{Z}_\hbar(k) = \mathbf{E}_i^T(k)\mathbf{W}_\hbar + \rho_\hbar, \ \ \hbar = 1, \ldots, n_f \tag{11}$$

$$\mathbf{H}_t(k) = \phi_t(\mathbf{Z}^f(k)\mathbf{W}_t + \rho_t), \ \ t = 1, \ldots, n_e \tag{12}$$

where $\mathbf{Z}^f(k) = [\mathbf{Z}_1(k), \ldots, \mathbf{Z}_{n_f}(k)]$ is the set of feature nodes. $\mathbf{W}_\hbar \in \mathbb{R}^{8\times n\hbar}$ and $\mathbf{W}_t \in \mathbb{R}^{(n\hbar\times n_f)\times m\hbar}$ are random weight matrices. $\rho_\hbar \in \mathbb{R}^{1\times n\hbar}$ and $\rho_t \in \mathbb{R}^{1\times m\hbar}$ are the random biases of $\mathbf{Z}_\hbar(k)$ and $\mathbf{H}_t(k)$. Besides that, $n\hbar$ and $m\hbar$ denote the numbers of $\mathbf{Z}_\hbar(k)$ and $\mathbf{H}_t(k)$ in one group, respectively.

Then, we build the activation function of the network. Let $\mathbf{H}^e(k) = [\mathbf{H}_1(k), \ldots, \mathbf{H}_{n_e}(k)]$ denotes the set of $\mathbf{H}_t(k)$. Combine $\mathbf{Z}^f(k)$ and $\mathbf{H}^e(k)$ into a new vector $\mathbf{S}_{n_e}(k) = [\mathbf{Z}^f(k)|\mathbf{H}^e(k)] = [\mathbf{Z}_1(k), \ldots, \mathbf{Z}_{n_f}(k)|\mathbf{H}_1(k), \ldots, \mathbf{H}_{n_e}(k)]$, which represents the activation function of BLS architecture.

Accordingly, one defines $\mathbf{S}_{n_e+1}(k)$ as

$$\mathbf{S}_{n_e+1}(k) = [\mathbf{S}_{n_e}(k)|\phi_t(\mathbf{Z}^f(k)\mathbf{W}_I + \rho_I)] \tag{13}$$

where $\mathbf{W}_I \in \mathbb{R}^{(n\hbar\times n_f)\times p}$ is a random weight matrix, and $\rho_I \in \mathbb{R}^{1\times p}$ denotes the random bias. $\mathbf{S}_{n_e+1}(k)$ is regarded as a new activation function vector with $p$ added enhancement nodes, which can be applied without retraining the whole network. In the following, a model-free formation algorithm with remouldable network structure is employed to explore the optimal update policy $\tau_i^*$. The BLS-based formation architecture of AUVs is presented in Fig. 1.

1) Initialization: Initially, $\tau_i^0(k)$ and $J_i^0(\zeta_i)$ are set to 0. $\mathbf{W}_\hbar$, $\mathbf{W}_t$, $\rho_\hbar$ and $\rho_t$ are randomly generated.

2) Network evaluation: Solve (10) by gradient descent with activation function $\mathbf{S}_{ne}(k)$. Define $J_i(\zeta_i(k))$ and $\tau_i^{s+1}(k)$ as

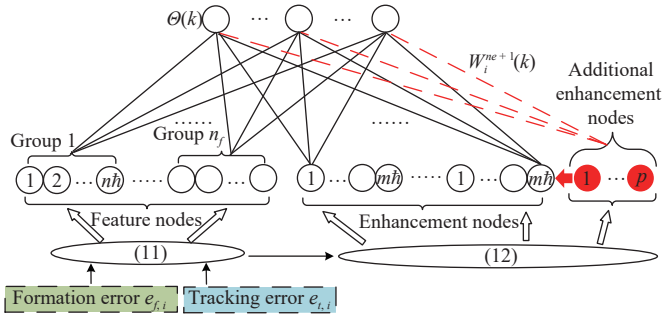



Fig. 1. Description of the BLS architecture.

$$J_i(\zeta_i(k)) = \mathbf{W}_{ne}^{c_i\,T}\mathbf{S}_{ne}^{c_i}(k)$$
$$\tau_i^{s+1}(k) = \left[\tau_{u_i}^{s+1}(k), \tau_{v_i}^{s+1}(k), \tau_{w_i}^{s+1}(k), \tau_{r_i}^{s+1}(k)\right]^T \tag{14}$$

where $\mathbf{S}_{n_e}^{c_i}(k)$ is the BLS-based basis function for the weight vector $\mathbf{W}_{c_i}$. Meanwhile, $\tau_\wp^{s+1}(k) = \mathbf{W}_{n_e}^{\wp\,T}\mathbf{S}_{ne}^{\wp}(k)$, where $\mathbf{S}_{ne}^{\wp}(k)$ is the BLS-based basis function for the weight vector $\mathbf{W}_\wp$. Based on this and noting with (10), one has

$$\sigma(k) = \mathbf{W}_{n_e}^{c_i\,T}(\mathbf{S}_{ne}^{c_i}(k+1) - \mathbf{S}_{ne}^{c_i}(k)) + g_i\left(\zeta_i(k), \tau_i^s(k)\right) + 2\sum\nolimits_{\wp\in\Omega}\mathbf{W}_{ne}^{\wp\,T}\mathbf{S}_{ne}^{\wp}(k)r_\wp\left(\tau_\wp(k) - \tau_\wp^s(k)\right) \tag{15}$$

where $\sigma(k)$ denotes the residual error. Let $\bar{\mathbf{S}}_{ne}(k) = [\Delta\mathbf{S}_{ne}^{c_i}(k); \mathbf{S}_{ne}^{u_i}(k); \mathbf{S}_{ne}^{v_i}(k); \mathbf{S}_{ne}^{w_i}(k); \mathbf{S}_{ne}^{r_i}(k)]$, $\Delta\mathbf{S}_{ne}^{c_i}(k) = \mathbf{S}_{ne}^{c_i}(k+1) - \mathbf{S}_{ne}^{c_i}(k)$, and $\mathbf{W}_i^{ne}(k) = [\mathbf{W}_{ne}^{c_i}; \mathbf{W}_{ne}^{u_i}; \mathbf{W}_{ne}^{v_i}; \mathbf{W}_{ne}^{w_i}; \mathbf{W}_{ne}^{r_i}]$. To minimize $\sigma(k)$, the weight update method can be obtained by the gradient descent method. In addition, $0 < \alpha < 1$ is the learning rate of the gradient descent method.

3) Network remodeling: In order to obtain better control fitting ability, the new activation function vector $\mathbf{S}_{ne+1}(k)$ is generated by (13). Define $\bar{\mathbf{S}}_{ne+1}(k)$ as $\bar{\mathbf{S}}_{ne+1}(k) = [\Delta\mathbf{S}_{ne+1}^{c_i}(k); \mathbf{S}_{ne+1}^{u_i}(k); \mathbf{S}_{ne+1}^{v_i}(k); \mathbf{S}_{ne+1}^{w_i}(k); \mathbf{S}_{ne+1}^{r_i}(k)]$. Accordingly, the new weight is updated as

$$\mathbf{W}_i^{ne+1}(k) = -\begin{bmatrix} \mathbf{W}_i^{ne}(k) - \mathbf{d}\mathbf{b}^T\Theta(k) \\ \mathbf{b}^T\Theta(k) \end{bmatrix} \tag{16}$$

with $\mathbf{b} = \begin{cases} (\mathbf{c})^+, & \text{if } \mathbf{c} \neq \mathbf{0} \\ (\mathbf{1} + \mathbf{d}^T\mathbf{d})^{-1}\mathbf{d}^T(\bar{\mathbf{S}}_{ne}(k))^+ & \text{if } \mathbf{c} = \mathbf{0} \end{cases}$, where $\mathbf{d} = (\bar{\mathbf{S}}_{ne}(k))^+ \times \phi_t(\mathbf{Z}^f(k)\mathbf{W}_I + \rho_I)$ and $\mathbf{c} = \phi_t(\mathbf{Z}^f(k)\mathbf{W}_I + \rho_I) - \bar{\mathbf{S}}_{ne}(k)\mathbf{d}$.

Theorem 1: The learning algorithm is convergent, provided that the learning rate $\alpha$ meets the following conditions:

$$0 < \alpha < \frac{2}{\max_\Lambda \max_k \|\bar{\mathbf{S}}_\Lambda(k)\|^2} \tag{17}$$

where $\Lambda \in \{ne, ne+1\}$.

Proof: Define the following Lyapunov function:

$$V(k) = \frac{1}{2}\sigma^2(k). \tag{18}$$

The difference of Lyapunov function is as follows:

$$\Delta V(k) = \frac{1}{2}\Delta\sigma(k)(2\sigma(k) + \Delta\sigma(k)) \tag{19}$$

where $\Delta\sigma(k) = \sigma(k+1) - \sigma(k)$. According to the approximate form of total differential, we have

$$\Delta\sigma(k) = \Delta\mathbf{W}_i^\Lambda(k)^T\frac{\partial P(k)}{\partial \mathbf{W}_i^\Lambda(k)} \tag{20}$$

where $P(k) = \mathbf{W}_\Lambda^{c_i\,T}\left(\mathbf{S}_\Lambda^{c_i}(k+1) - \mathbf{S}_\Lambda^{c_i}(k)\right) + 2\sum_\wp \mathbf{W}_\Lambda^{\wp\,T}\mathbf{S}_\Lambda^\wp(k)\ r_\wp\left(\tau_\wp(k) - \tau_\wp^s(k)\right)$, and $\Delta\mathbf{W}_i^\Lambda(k) = \mathbf{W}_i^\Lambda(k+1) - \mathbf{W}_i^\Lambda(k)$.

Next, we can get $\Delta\mathbf{W}_i^\Lambda(k) = -\alpha\frac{\partial V(k)}{\partial \mathbf{W}_i^\Lambda(k)} = -\alpha\sigma(k)\frac{\partial P(k)}{\partial \mathbf{W}_i^\Lambda(k)}$. Finally, we can deduce

$$\Delta V(k) = -\frac{1}{2}\alpha\sigma^2(k)\left\|\bar{\mathbf{S}}_\Lambda(k)\right\|^2\left(2 - \alpha\left\|\bar{\mathbf{S}}_\Lambda(k)\right\|^2\right). \tag{21}$$

According to the Lyapunov stability theorem, the asymptotic stability is guaranteed if $\Delta V(k) < 0$. ■

**Simulation and experiment results:** In the simulation, we consider that four AUVs ($N = 4$) require to achieve the formation task, where AUV 1 can communicate with target. Meanwhile, $a_{12} = a_{14} = a_{32} = a_{34} = 1$ and $a_{13} = a_{24} = 0$. The initial states of AUVs are $\zeta_1(0) = [0, 0, -15, 1, 0, 0, 0, 0]^T$, $\zeta_2(0) = [0, 20, -15, 1, 0, 0, 0, 0]^T$, $\zeta_3(0) = [20, 20, -15, 1, 0, 0, 0, 0]^T$ and $\zeta_4(0) = [20, 0, -15, 1, 0, 0, 0, 0]^T$. The state of target is $\zeta_r = [6, 6, 0, 0, 0, 0, 0, 0]^T$. In the formation process, the desired shape is a square whose side length is 6. Accordingly, the formation trajectories of AUVs are provided in Fig. 2(a). Clearly, AUVs move to the centre and also to the target. Meanwhile, $d_{12}$, $d_{14}$, $d_{32}$ and $d_{34}$ denote the relative distances between AUVs, which are shown in Fig. 2(b). The above results are very close to the desired distances, i.e., 6 m. Fig. 3 (a) shows the position and angle errors of AUVs. It demonstrates that the formation controller in this

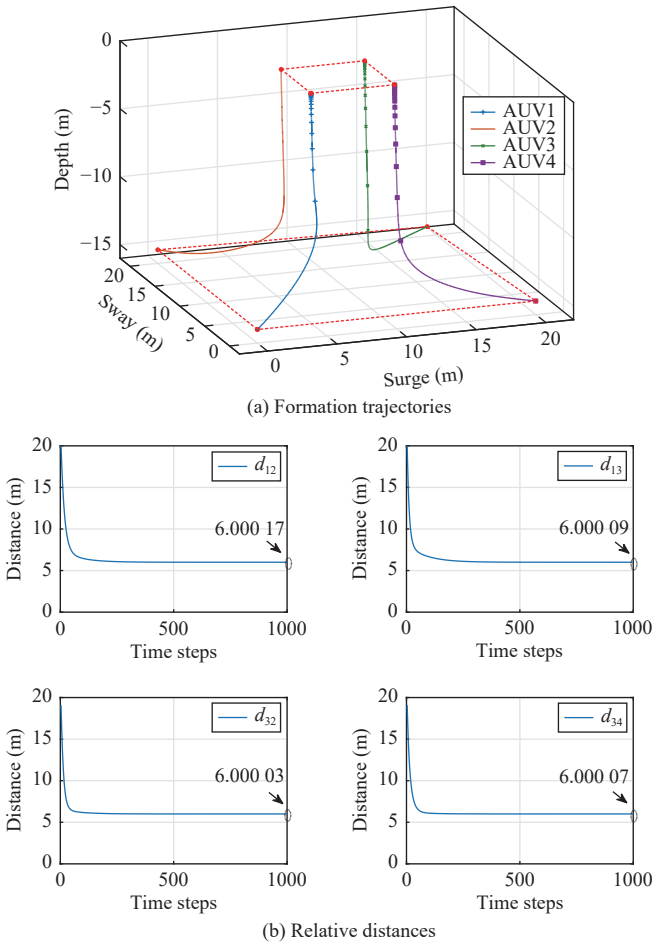



Fig. 2. Formation trajectories and relative distances in incremental learning.

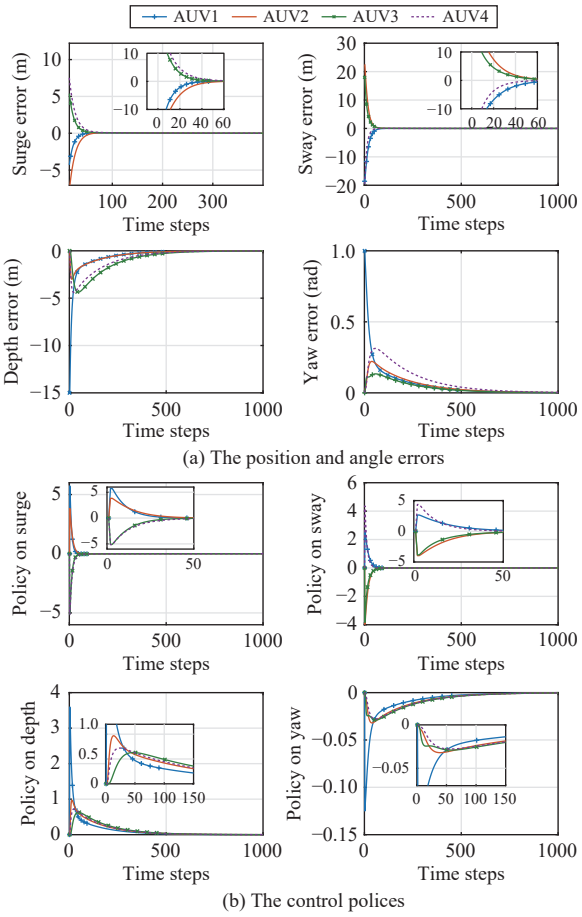



Fig. 3. The state errors and control polices in incremental learning.

letter can make the state errors of AUVs converge to 0 and achieve the formation task. Fig. 3(b) shows the value of control policy during formation. From Figs. 2 and 3, the incremental learning-based formation algorithm takes 520 time steps to achieve convergence.

One important property of the our solution is to remodel the learning framework without a retraining process. To verity its merit, the following two scenarios are considered: 1) Remodel the learning framework with additional enhancement nodes, i.e., the solution in this letter; 2) The learning framework is fixed, e.g., the solutions in [8]. The results in Scenario 2 are shown in Fig. 4. Similar to the trajectories in Fig. 2, one knows that the formation task in Scenario 2 can be achieved. Nevertheless, the training time in Scenario 2 is longer than the results in Scenario 1, which results in the slow convergence of formation, i.e., 1000 time steps. Of note, the training time is reduced with the number increase of enhancement nodes.

Another important characteristic of our formation solution is that the control performance is better than the DL method under the same weight calculation amount. To verity its merit, similar to [12], the DL-based method is employed to achieve the formation task of AUVs, where the fully connected neural network structure for each degree is designed as the form of $3\times8\times3$. Accordingly, the calculation amount of single network weight is given as 48, which is the same as the proposed algorithm. Based on this, the simulation results are shown in Fig. 5. Clearly, the formation performance by using BLS is much better than the one with DL.

The field experiment is conducted. The hardware structure of the experimental platform is shown in Fig. 6(a). Due to the limited conditions in our lab, AUVs are not equipped with underwater acoustic communicator. Thus, AUVs cannot acquire the accurate position information when they debry flow under water. In view of this, the experimental results are conducted on the water surface. In our future work, we will equip the underwater communication unit into AUVs,

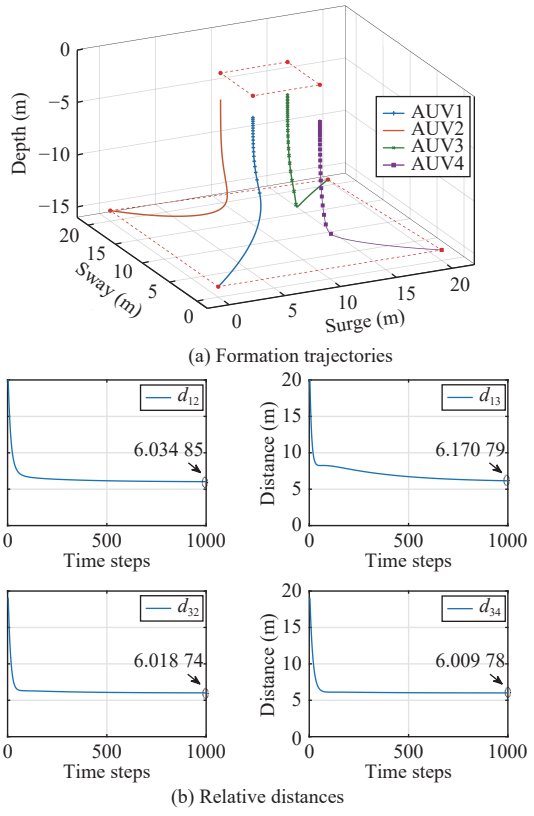
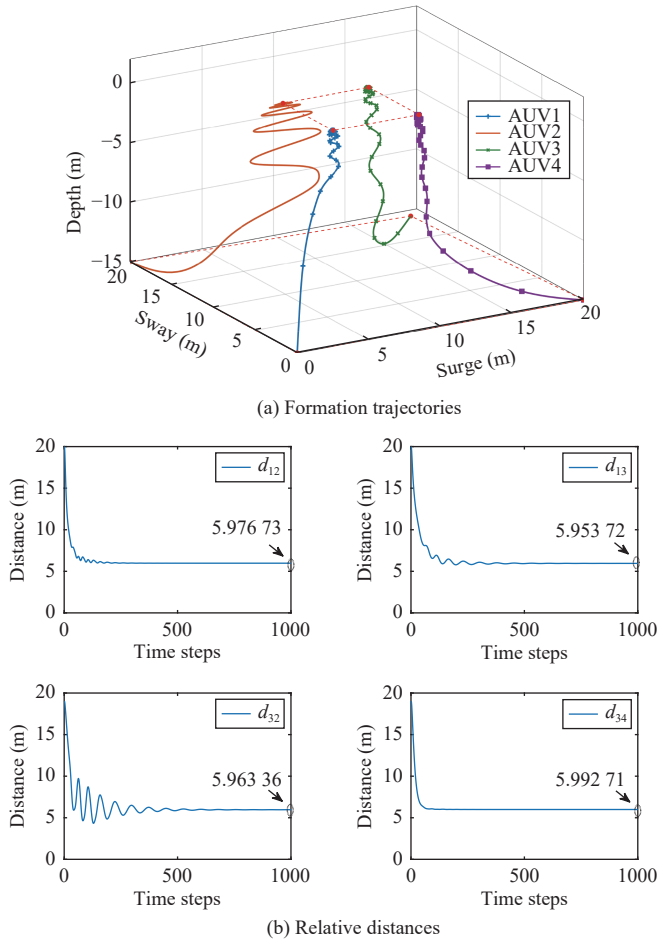


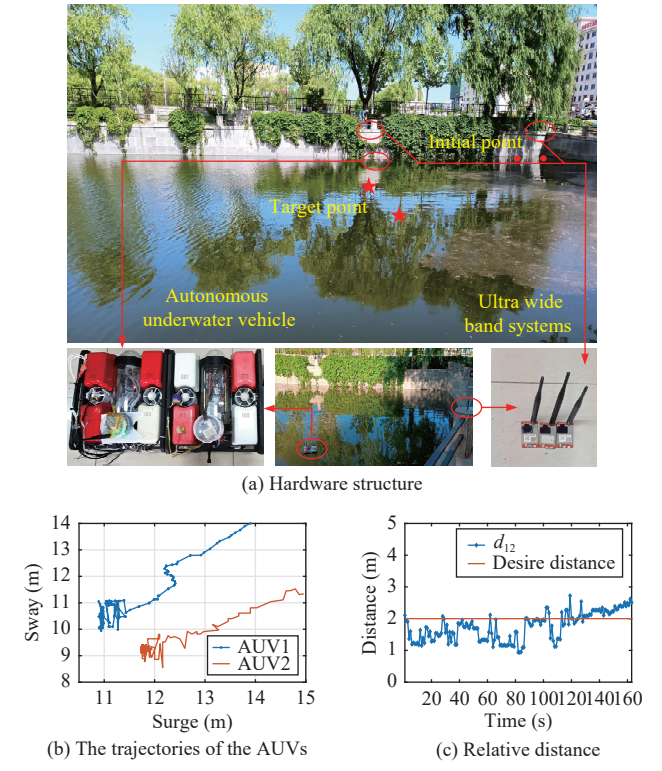

Fig. 4. The formation trajectories and relative distances in fixed network.




(a) Formation trajectories

(b) Relative distances

Fig. 5. The formation trajectories and relative distances in deep learning.




(a) Hardware structure

(b) The trajectories of the AUVs

(c) Relative distance

Fig. 6. Experimental results in the Yanming Lake.

such that underwater experiments can be conducted. The initial positions of the two AUVs are $\zeta_1(0) = [10.8, 9.9, 0, 0, 0, 0, 0, 0]^T$, $\zeta_2(0) = [11.8, 9.2, 0, 0, 0, 0, 0, 0]^T$, respectively. The target position is $\zeta_r(0) = [14, 14, 0, 0, 0, 0, 0, 0]^T$. The relative distance between two AUVs is 2 m. The relative position vector is $\mathbf{r}_{12} = [1.41, -1.41]^T$. In Fig. 6(b) and 6(c), the trajectories and distance error of the AUVs are shown. It can be seen that AUVs basically maintain formation shape. The error between the actual distance and the desired distance is within the allowable range. The video of the formation experiment is given by https://v.youku.com/v_show/id_XNTg4Njk0NDM3Mg==.html.

**Conclusion and future work:** In this letter, a model-free incremental learning formation algorithm is proposed for AUVs. Specifically, the BLS-based structure can simplify the deep learning network framework and reduce the training time. Finally, the simulation and field experiment results illustrate the effectiveness of our algorithm. In future, we will investigate the influences of environmental disturbances and unmodeled hydro-dynamics on the formation of AUVs.

**Acknowledgments:** This work was supported in part by the National Natural Science Foundation of China (62222314, 61973263, 61873345, 62033011); the Youth Talent Program of Hebei (BJ2020031); the Distinguished Young Foundation of Hebei Province (F2022203001); the Central Guidance Local Foundation of Hebei Province (226Z3201G); and the Three-Three-Three Foundation of Hebei Province (C20221019).

## References

[1] L. Ma, Y. Wang, and Q.-L. Han, " Cooperative target tracking of multiple autonomous surface vehicles under switching interaction topologies," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 673–684, Mar. 2023.

[2] Y. Chen and P. Wei, " Coordinated adaptive control for coordinated path-following surface vessels with a time-invariant orbital velocity," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 4, pp. 337–346, Oct. 2014.

[3] Z. Gao and G. Guo, " Fixed-time sliding mode formation control of AUVs based on a disturbance observer," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 539–545, Mar. 2020.

[4] Y. Zhao, Y. Ma, and S. Hu, "USV formation and path-following control via deep reinforcement learning with random braking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5468–5478, 2021.

[5] Y. Shou, B. Xu, A. Zhang, and T. Mei, " Virtual guidance-based coordinated tracking control of multi-autonomous underwater vehicles using composite ceural learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5565–5574, Dec. 2021.

[6] C. Philip and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[7] L. Yuan, T. Li, C. Philip, Q. Shan, and M. Han, " Broad learning systembased learning controller for course control of marine vessels," in *Proc. 10th Int. Conf. Intell. Control Inf.*, 2019, pp. 133–136.

[8] X. Gao, W. Bai, T. Li, L. Yuan, and Y. Long, " Broad learning systembased adaptive optimal control design for dynamic positioning of marine vessels," *Nonlinear Dyn.*, vol. 105, no. 1, pp. 1593–1609, 2021.

[9] R. Cui, C. Yang, Y. Li, and S. Sharma, " Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 6, pp. 1019–1029, Jun. 2017.

[10] J. Yan, X. Li, X. Luo, C. Hua, and X. Guan, "Integrated localization and tracking for AUV with model uncertainties via scalable sampling-based reinforcement learning approach," *IEEE Trans. Syst. Man Cybern.*, 2021. DOI: 10.1109/TSMC.2021.3129534

[11] X. Sun and S. Ge, "Adaptive neural region tracking control of multifully actuated ocean surface vessels," *IEEE/CAA J. Autom. Sinica*, vol. 1, no. 1, pp. 77–83, 2014.

[12] J. Hao, G. Zhang, W. Liu, Y. Zheng, and L. Ren, "Data-driven tracking control based on LM and PID neural network with relay feedback for discrete nonlinear systems," *IEEE Trans. Ind. Electron.*, vol. 68, no. 11, pp. 11587–11597, Nov. 2021.