Communications Engineering

https://doi.org/10.1038/s44172-025-00571-7

Article in Press

Digital twin-driven swarm of autonomous underwater vehicles for marine exploration

Received: 29 July 2025

Accepted: 3 December 2025

Cite this article as: Yan, J., Zhang, T., Guan, X. *et al.* Digital twin-driven swarm of autonomous underwater vehicles for marine exploration. *Commun Eng* (2025). https://doi.org/10.1038/s44172-025-00571-7

Jing Yan, Tianyi Zhang, Xinping Guan, Xian Yang & Cailian Chen

Digital Twin-Driven Swarm of Autonomous Underwater Vehicles for Marine Exploration

Jing Yan[+,*], Tianyi Zhang[+,*], Xinping Guan[*], Xian Yang, and Cailian Chen

November 6, 2025

Abstract

Swarm control of autonomous underwater vehicles (AUVs) has been recognized as the foundation for marine exploration. However, the implementation of this task faces two major constraints: excessive communication energy demands and limited environmental perception capabilities. This article proposes a digital twin (DT)-driven swarm control of AUVs solution to overcome these limitations. We first create the digital replicas for each AUV by integrating the dynamics and environmental data. With the collected states from AUVs, a parameter estimator is proposed to predict the flow field, while a swarm networking protocol is designed to reduce the energy consumption. After that, an integral reinforcement learning (IRL)-based swarm controller is proposed to drive the virtual and real AUVs. Based on the interaction information between DT models and AUVs, the virtual-real error optimization algorithm is implemented to minimize the matching errors. Finally, the effectiveness of our solution is verified by the experimental results. These results demonstrate that the DT-driven swarm control of AUVs can improve the underwater situation awareness and prediction accuracy while reducing the communication energy consumption.

Introduction

The ocean holds vast resources critical for scientific, industrial, and environmental applications, such as deep-sea exploration, offshore infrastructure maintenance, and ecological monitoring[1-3]. To accomplish these underwater tasks, autonomous underwater vehicles (AUVs) are widely deployed due to their unmanned and adaptive capabilities[4, 5]. However, the individual AUV faces limitations in operational range and efficiency. The swarm of AUVs can address these challenges by enabling collaborative formations, which enhance sensing coverage, task robustness, and mission flexibility. Particularly, the swarm control ensures that multiple AUVs maintain a predefined geometric shape while navigating dynamic underwater environments[6, 7], making it a cornerstone for advanced marine exploring.

To date, many swarm control methods have been proposed in the field of AUVs operations. These methods can be roughly divided into the following categories: bio-inspired behavior-based swarms[8, 9], mathematical model-based swarms[10, 11], dynamically reconfigurable swarms[12, 13], and task-driven collaborative swarms[14, 15]. Specifically, the bio-inspired behavior-based swarms mimic fish/bird group behavior, utilizing decentralized self-organization principles for adaptive swarm control; the mathematical model-based swarms employ formal frameworks including virtual structures, graph theory, and artificial potential fields to achieve precise swarm control; the dynamically reconfigurable swarms utilize algorithms like finite-state machines or reinforcement learning, enabling real-time and seamless swarm transitions; the task-driven collaborative

*J. Yan, T.Zhang, and X. Yang are with the Department of Electrical Engineering, Hebei Key Laboratory of Industrial Computer Control Engineering, and also with the Key Laboratory of Intelligent Control and Neural Information Processing, Ministry of Education, Yanshan University, Qinhuangdao 066004, China; X. Guan and C. Chen are with the Department of Automation, Shanghai JiaoTong University, and also with the State Key Laboratory of Submarine Geoscience, Shanghai 200240, China; + The two authors contribute equally to this work. * Corresponding authors. Email: jyan@ysu.edu.cn, zt2001@stumail.ysu.edu.cn, xpguan@sjtu.edu.cn

swarms assign heterogeneous AUVs to undertake different functions (sensing, communication relay, etc.) with synchronized coordination to optimize swarm missions. However, these swarm methods[8-15] rely heavily on real-time interaction communication. As the primary means of data exchange, the underwater acoustic communication consumes significant energy and suffers from latency[16], making continuous communication unreasonable. In order to reduce reliance on real-time data, a computational fluid dynamics-based method was proposed in ref. 17 for predicting behaviors. In ref. 18, a variable fuzzy predictor was developed to accurately estimate the motion of AUVs. Besides, our previous work[19] designed a tracking algorithm that utilizes mobility prediction for AUV, ensuring both control stability and energy efficiency. Although the aforementioned works[17-19] can improve the energy efficiency of AUVs, these control strategies are prone to accumulate errors over time. At the same time, underwater environments are also more difficult to perceive compared to terrestrial. For example, high turbidity significantly degrades optical clarity, severely limiting the effectiveness of camera-based vision systems for long-range swarm control[20]. Although the sonar can extend detection range compared to optical methods, they are constrained to provide only two-dimensional environmental data. To the best of our knowledge, how to improve the environmental awareness and prediction precision while minimizing communication demands becomes the primary issue to resolve.

With the above issue in hand, we notice that the digital twin (DT) technology emerges as a transformative solution by creating virtual replicas of physical AUVs and their environment[21]. Particularly, DT can be defined as virtual, real-time replicas of physical systems or objects, created using sensor data and computer models to optimize performance and predict issues[22]. For example, a teleoperation control framework incorporating DT technology was developed in ref. 23, which can improve environmental perception through intuitive visualization. In ref. 24, a DT-based simulation platform was designed to evaluate the performance of diver propulsion vehicle across various operational conditions. Nevertheless, these works[23, 24] fail to establish an effective optimization mechanism between virtual models and physical entities, potentially leading to significant error propagation. Along with this, the depth deterministic policy gradient with particle swarm optimization was combined in ref. 25 to minimize model matching errors. In ref. 26, a value-based reinforcement learning algorithm was developed for virtual-real synchronization. Similarly, in order to enhance model optimization accuracy, a DT system incorporating multi-disciplinary optimization solution was implemented in ref. 27. Although these methods[25-29] demonstrate effectiveness in single-vehicle tracking scenarios, they cannot be directly applied to the swarm control due to the strong coupling relationship of AUVs. For that reason, the previous work[30] proposed the DT-driven formation control of underwater vehicles, and a matching error optimization algorithm was designed to minimize the virtual-real error. However, it ignores the impact of ocean current disturbances on underwater vehicles, resulting in unstable swarm control of AUVs. Meanwhile, some flow field estimating methods[31-33] were not consider ocean disturbances in the heave direction, which result in incomplete underwater situational awareness and inaccurate information prediction. In addition, to ensure operational safety of AUVs, the obstacle avoidance (including neighboring vehicles and environmental hazards) is important. However, these factors were not adequately addressed in the aforementioned works. To sum up, how to design a DT-driven swarm controller, which combines flow field estimation and collision avoidance remains an unresolved challenge.

Here, we propose the DT-driven swarm of AUVs control solution for marine exploring, as depicted by Fig. 1. Specifically, we first construct the DT models, which include the dynamic model of AUVs and the flow field model. Based on this, the Newton iteration-based parameter estimator is proposed to predict the three-dimensional (3D) flow field. After that, the integral reinforcement learning (IRL)-based swarm controller is proposed to drive the virtual and real AUVs, while the virtual-real error optimization algorithm is implemented to minimize the matching errors between DT models and AUVs. Finally, the effectiveness of our solution is verified by the experimental results. Compared with the existing works, the main contributions of our solution can be provided as follows: 1) *Construction of the DT-driven architecture.* The proposed DT-driven architecture comprises four core components: the virtual replication module that digitally mirrors AUVs and environment, the prediction of 3D flow field, the energy-efficient swarm networking protocol for data exchange communication, and the real AUVs for swarm control. This architecture offers distinct advantages over the existing works: it reduces the communication consumption compared to conventional

real-time interaction swarm systems[8-15], while enhancing underwater situation awareness and prediction accuracy beyond the capabilities of mobility prediction-based works[17-19]. 2) *Design of the swarm controller and optimization algorithm.* We develop an IRL-based swarm control system, which includes the swarm controller ensuring swarm stability and an IRL-based optimization module minimizing matching error. This approach fundamentally advances the current DT systems[23, 24] by incorporating optimization mechanisms that prevent error accumulation. Compared with the single-vehicle solutions[25-29], our architecture addresses the challenges of multiple AUVs coordination while incorporating essential collision avoidance capabilities. Different from the previous work[30] and planar estimation[31-33], the proposed swarm controller considers the impact of ocean current disturbances and can predict the 3D real-time flow field.

Results

Experiment in the pool environment

As depicted in Fig. 2(a), the hardware system comprises several key components: 1) *AUVs.* The swarm control task is implemented using four AUVs equipped with STM32 microcontrollers and NVIDIA processors, with AUV 1 designated as the swarm leader; 2) *Communication module.* Each AUV is equipped with NRF24L01 transceivers for data exchange, employing radio frequency transmission for surface operations and acoustic communication for underwater environments; 3) *Positioning module.* Dual localization capabilities are provided through integrated ultra-wideband (UWB) and ultra-short baseline (USBL) technologies. For the DT platform development, a combination of Blender, Unity, and Visual Studio software packages are employed, as illustrated in Fig. 2(b).

For the mentioned above platform, the depth sensors, inertial motion unit (IMU), acoustic doppler current profiler (ADCP), and vision sensors are integrated into the swarm control system, whose structural relationship is depicted by Fig. 3. Particularly, the STM32 and NVIDIA processors are used for low-level control and high-level computation. The depth sensors obtain depth information of AUVs, such that the positions of AUVs can be measured by the assistance of USBL and UWB. The vision sensors provide image information of AUVs and environment, whose aim is to reconstruct DT models and conduct environmental perception. The IMU can provide linear acceleration and yaw angle of AUVs, which is used to correct the attitude information. To better characterize the underwater environment disturbances, the ADCP can be employed to measure the current information.

Effectiveness of the DT-driven swarm controller

For this experiment conducted in the pool environment, the true flow field at $\mathbf{p}_c = [1, 4.6, -0.2]^{\mathrm{T}}$ m can be obtained using the ADCP, with $\mathbf{v}^c = [-1.7, 5.2, -3.8]^{\mathrm{T}}$ cm/s. For the swarm of AUVs, the destination is defined as $\mathbf{p}_t = [-0.9, 5.9, -0.1]^{\mathrm{T}}$ m, while the expected relative distances are $\boldsymbol{\gamma}_{12} = [1.8, 0, 0]^{\mathrm{T}}$ m, $\boldsymbol{\gamma}_{23} = [-2.4, -0.5, -0.7]^{\mathrm{T}}$ m, $\boldsymbol{\gamma}_{34} = [3, 0, 0]^{\mathrm{T}}$ m and $\boldsymbol{\gamma}_{41} = [-2.4, 0.5, 0.7]^{\mathrm{T}}$ m. Besides, two static cylindrical obstacles are modeled in the environment, with their base centers located at $\mathbf{O}_1 = [1.5, 4, -1]^{\mathrm{T}}$ m and $\mathbf{O}_2 = [-1.5, 4, -1]^{\mathrm{T}}$ m, and their respective heights specified as $h_1 = 0.7$ m and $h_2 = 1.4$ m. Some other parameters used in the experiment are presented in Table 1.

We first validate the Newton iteration-based parameter estimator. Specifically, the swarm of AUVs operating environment is treated as an unknown flow field to be predicted. Fig. 2(c) shows the true flow field distribution, where the optimal flow field parameter is denoted as α^*. By substituting the iteratively estimated parameters into equation (2), the predicted flow field is obtained, as illustrated in Fig. 2(d). Furthermore, the prediction error between the predicted and true flow fields is defined as: $e_c = \mathrm{norm}(\mathbf{v}_i^c(\alpha^*) - \mathbf{v}_i^c(\alpha))$. As shown in Fig. 2(e), the error distribution demonstrates that the prediction error is on the order of 10^{-3} m/s, which meets the accuracy requirements for flow field prediction in real-world experimental scenarios.

With the developed swarm controller, the optimal control policies τ_i^* and $\hat{\tau}_i^*$ can be obtained to drive the swarm of AUVs. According to τ_i^* and $\hat{\tau}_i^*$, the swarm trajectories of AUVs and DT models are depicted in Fig.

Table 1: Main parameters used in the experiment.

Parameter	Value	Parameter	Value	Parameter	Value
$\overline{\alpha}$	0.5	λ	1	M	6
μ	1	δ_α	0.3	β	0/1
$\rho_{io}^{\overline{m}}$	0.25m	δ_V	0.3	δ_τ	0.4
k_{ij}	0.4	$\delta_{V_{ei}}$	0.5	$R_\#$	0.2m
$k_{o,i}$	0.6	δ_{ϕ_i}	0.6	H	0.15m
T_b	10^3bps	E_elec	50^{-9}kJ/bit	$P(\bar{f})$	0.035
T_w	5s	$\bar{f}$	10Hz	C	$4.2 * 10^{-9.5}$

2(f) and Fig. 2(g), respectively. These results can demonstrate the DT models can restore the states of AUVs. For quantitative evaluation, we define the swarm error metric $\mathbf{e}_{ij}$ as $E_{ij} = \|\mathbf{e}_{ij}\|$ (where $i, j \in \{1, 2, 3, 4\}$, $i \neq j$), with the norm of swarm error vectors shown in Fig. 2(h). The asymptotic convergence of swarm errors to zero confirms successful swarm maintenance.

Next, the safety performance is validated through obstacle avoidance analysis (Fig. 2(i)) and relative distances between any two AUVs monitoring (Fig. 2(j)), where all measured distances consistently exceed the minimum safety threshold, i.e., 0.25 m. Furthermore, the state matching errors between virtual-real AUVs, quantified by $E_i = \|\mathbf{e}_i\|$ (Fig. 2(k)), verify the optimization algorithm. These experimental results demonstrate the effectiveness of the DT-driven swarm controller.

Comparison with other swarm controllers

A critical advantage of our proposed method lies in its reduction of communication energy consumption. To validate this judgement, we conduct comparative experiments by eliminating the DT models from the swarm control system, adopting a configuration similar to those works in refs. 8-15. Referring to ref. 30, the total energy consumption of transmitting and receiving m bits with distance l can be represented as $E_\mathrm{total}(m, l) = 2mE_\mathrm{elec} + mT_\mathrm{b}CHe^{P(\bar{f})l}$, where T_b, E_elec, $\bar{f}$, $P(\bar{f})$, H, and C are bit duration, energy consumption of processing one bit, channel center frequency, absorption coefficient, depth and coefficient, respectively. Based on the above theoretical analysis, the experimental results of energy consumption comparison are presented in Fig. 4(a) and Fig. 4(b), respectively. We can find that the conventional approaches accomplish swarm control, but they incur higher energy consumption owing to stringent real-time communication demands.

Furthermore, our solution enhances environmental adaptability through the optimization algorithm. To evaluate this capability, we design a changing test scenario involving a mass reduction of AUV 3 and AUV 4, while raising their operating depth from -0.8 m to -0.2 m. Next, we implement a DT-driven swarm system without the matching algorithm (analogous to refs. 23, 24), where the trajectories of AUVs and DT models are depicted in Fig. 4(c) and Fig. 4(d), respectively. By contrast, we implement our proposed swarm control strategy under the identical environmental conditions. The corresponding trajectories for both physical AUVs and their DT models are illustrated in Fig. 4(e) and Fig. 4(f), respectively. Specifically, the swarm error $\mathbf{e}_{ij}$ and virtual-real error $\mathbf{e}_i$ are regarded as the quantitative indicators. Based on the above indicators, the swarm error is 0.8 m and the corresponding virtual-real error ultimately reaches 0.6 m in Figs. 4(c)-(d), while the swarm error and virtual-real error converge to zero in Figs. 4(e)-(f). It means that the swarm system without the matching algorithm exhibits divergence with progressively increasing matching errors, ultimately leading to swarm failure. However, the proposed matching algorithm in our solution can achieve the successful execution of swarm maintenance.

Experiment in the lake environment

Effectiveness of the DT-driven swarm controller

To validate the proposed method's practical applicability, the field experiments are conducted in the

lake, where the experimental configurations are illustrated in Fig. 5(a) and Fig. 5(b). Specifically, the actual current velocity field is measured at position $\mathbf{p}_c = [3.5, 4.1, -0.3]^T$ m, with a flow velocity of $\mathbf{v}^c = [0.1, -0.12, -0.08]^T$ m/s. The destination for the AUV swarm is established at $\mathbf{p}_t = [-0.2, 7.2, -0.1]^T$ m, with the following expected swarm parameters: $\boldsymbol{\gamma}_{12} = [0.4, 0, 0]^T$ m, $\boldsymbol{\gamma}_{23} = [-0.8, -0.2, -0.3]^T$ m, $\boldsymbol{\gamma}_{34} = [1.2, 0, 0]^T$ m and $\boldsymbol{\gamma}_{41} = [-0.8, 0.2, 0.3]^T$ m. Different from the static obstacles in the pool, we adopt a dynamic cylindrical obstacle in the lake, with its center is $\mathbf{O}_1 = [0, 4, -1]^T$ m and its height is $h_1 = 1.4$ m.

The actual flow field distribution is depicted in Fig. 5(c). Through the iteratively estimated parameters, the predicted flow field can be shown in Fig. 5(d). Next, the prediction error e_c between the estimated and actual flow fields can be presented in Fig. 5(e). From these results, we can find the prediction error is about 10^{-3} m/s, which meets the accuracy requirements for flow field prediction in real underwater experimental scenarios.

Furthermore, the swarm of AUVs is driven using the IRL-based swarm controller. The resulting trajectories for both physical AUVs and their DT models are presented in Fig. 5(f) and Fig. 5(g), respectively. The results demonstrate the DT models' capability to accurately replicate the AUVs' dynamic states. At the same time, the quantitative evaluation of swarm maintenance is provided in Fig. 5(h), where the asymptotic convergence of swarm errors to zero confirms successful task completion. After that, the safety performance is assessed through two key metrics: AUVs-obstacles relative distances (Fig. 5(i)) and inter-AUVs relative distances (Fig. 5(j)). Both measurements consistently exceed the predefined safety thresholds and verify collision-free operation. Besides that, the norm of matching errors between physical AUVs and DT models, quantified by Fig. 5(k), validates the matching error optimization algorithm. These experimental results demonstrate the effectiveness of the DT-driven swarm controller in the lake.

Comparison with other swarm controllers

To demonstrate the superiority of the DT-driven swarm architecture, we conduct the comparative experiments by removing the DT models from the swarm system, following approaches similar to those in refs. 8-15. As shown in Fig. 6(a), the experimental results show the trajectories of AUVs based solely on swarm errors from physical units. Although the swarm task can still be accomplished, Fig. 6(b) reveals significantly higher communication energy consumption compared to the developed DT-driven architecture.

Furthermore, we investigate the environmental adaptability of our solution under changing conditions. Specifically, we design an underwater scenario where the mass of AUVs 3 and 4 decreased, and their operating depth change from -0.4 m to -0.2 m.When the matching algorithm is removed from the DT-driven architecture (as in refs. 23, 24), the swarm control degraded noticeably. Fig. 6(c) and Fig. 6(d) demonstrate the resulting divergent AUVs' trajectories and growing matching errors, ultimately causing swarm failure. Although its swarm error $\mathbf{e}_{ij}$ converges to zero, the corresponding virtual-real error $\mathbf{e}_i$ ultimately reaches 0.4 m. In contrast, when applying our complete swarm solution under the same conditions, the system maintains stable performance. As evidenced by Fig. 6(e) and Fig. 6(f), the physical AUVs and their DT models both achieve and maintain the desired swarm, where the swarm error and the virtual-real error eventually approach zero. These comparative experiments clearly validate the effectiveness and advantages of our proposed method in this article.

Experiment in the near-shore of sea

Effectiveness of the DT-driven swarm controller

In order to further demonstrate the practicality, we also validate the effectiveness in the near-shore of sea, where the experiment setup can be shown in Fig. 7(a) and Fig. 7(b). Specifically, the flow velocity at location $\mathbf{p}_c = [4.5, 5.3, -0.4]^T$ m is measured by an ADCP, with the recorded flow velocity being $\mathbf{v}^c = [-0.2, -0.28, -0.15]^T$ m/s. For the swarm of AUVs, the destination is set as $\mathbf{p}_t = [-0.6, 7.0, -0.1]^T$ m, while the desired swarm position vectors are $\boldsymbol{\gamma}_{12} = [0.4, 0, 0]^T$ m, $\boldsymbol{\gamma}_{23} = [-1, -0.2, -0.3]^T$ m, $\boldsymbol{\gamma}_{34} = [1.6, 0, 0]^T$ m and $\boldsymbol{\gamma}_{41} = [-1, 0.2, 0.3]^T$ m. In addition, the underwater environment contains two dynamic cylindrical

obstacles for swarm testing. The first obstacle has its base centered at $\mathbf{O}_1 = [-1.2, 4.3, -1]^{\mathrm{T}}$ m with a height of 1.4 m, and the second obstacle is positioned at $\mathbf{O}_2 = [0.9, 3, -1]^{\mathrm{T}}$ m with the height of 1.4 m.

The actual flow field distribution (Fig. 7(c)) with the optimal parameter α^* serves as ground truth for verification. By applying our estimation method to equation (2), we obtain the predicted flow field shown in Fig. 7(d). As depicted in Fig. 7(e), the results demonstrate prediction error e_c at 5×10^{-3} m/s, satisfying the practical application requirements.

With the swarm controller, the optimal control policies τ_i^* and $\hat{\tau}_i^*$ guide the motion of AUVs. Fig. 7(f) and Fig. 7(g) present the trajectories of physical AUVs and their DT models respectively, validating the DT's ability to accurately replicate the motion states of physical AUVs. At the same time, the norm of swarm errors converges to zero (Fig. 7(h)), confirming the swarm maintenance capability. Furthermore, the safety validation included two aspects: the obstacles avoidance performance (Fig. 7(i)) and inter-AUVs distances monitoring (Fig. 7(j)). From these results, we can obtain that all measured distances remain above the 0.25 m safety threshold. Additionally, Fig. 7(k) shows the virtual-real state matching errors, which verifies the optimization algorithm's effectiveness. To sum up, these comprehensive results demonstrate the reliability of our proposed DT-driven swarm controller architecture in the complex sea environment.

Comparison with other swarm controllers

In order to evaluate the advantages of our DT-driven swarm control architecture, we perform comparative experiments by removing the DT models, as similar to those works in refs. 8-15. The trajectories of AUVs based solely on swarm errors from physical units are depicted in Fig. 8(a). Although the swarm tasks can still be achieved, the energy analysis in Fig. 8(b) shows that our DT-driven solution achieves a significant reduction in communication energy consumption compared to traditional methods.

After that, to assess the environmental adaptability of our proposed method, we conduct experiments under changing underwater conditions where the mass of AUVs 3 and 4 is reduced and their operating depth changes from -0.4 m to -0.2 m. When testing traditional methods without the matching algorithms (as implemented in refs. 23, 24), the system exhibits AUVs' trajectory divergence and cumulative matching errors as shown in Fig. 8(c) and Fig. 8(d), ultimately leading to swarm failure. By contrast, the physical AUVs and their DT models successfully achieve and maintain the desired swarm, as depicted in Fig. 8(e) and Fig. 8(f). Particularly, the swarm error $\mathbf{e}_{ij}$ is 0.6325 m and the virtual-real error $\mathbf{e}_i$ reaches 1.1 m in Figs. 8(c)-(d), however the swarm error and virtual real error with our solution can eventually approach the allowable range, as shown in Figs. 8(e)-(f). These experimental results provide clear evidence of the enhanced robustness and superior performance offered by our proposed DT-driven architecture in dynamic underwater environments.

Remark: In the experimental results, the static and dynamic obstacles are both considered. For example, the cylindrical tubes in the pool are static, and then they are considered as the static obstacles, while the neighboring AUVs are considered as dynamic obstacles. For the outdoor environment, the cylindrical tubes passively move with the current, and thereby, they are considered as dynamic obstacles in the lake and sea experiments. It should be pointed out that the procedure of dynamic obstacle avoidance is more complex, as compared with the static scenario.

Discussion

In this article, we study a DT-driven swarm of AUVs for marine exploring. To improve the underwater situation awareness and prediction accuracy, we construct a DT-driven swarm architecture of AUVs. Meanwhile, a parameter estimator is proposed to predict the flow field, while a swarm networking protocol is designed to reduce the energy consumption. After that, an IRL-based swarm controller is proposed to drive the virtual and real AUVs. According to the interaction information between DT models and AUVs, the virtual-real

error optimization algorithm is implemented to minimize the matching errors. Finally, the effectiveness of our solution is verified by the experimental results.

Our strategy is inspired by the basis idea in ref. 27, but it is fundamentally different from the solution in ref. 27. Particularly, the main differences can be summarized as follows: 1) *Overall design*. This article focuses on the swarm control of AUVs based on DT technology to perform underwater tasks, while the development and validation of individual underwater vehicles are studied in ref. 27; 2) *3D flow field prediction*. Although the influence of ocean current is considered in ref. 27, no corresponding description is given. This article presents a three-dimensional flow field model (see equation (2)) and predicts the flow field; 3) *Swarm networking protocol*. To further reduce the communication energy consumption, this article proposes the swarm networking protocol for AUVs. However, this protocol was not considered in ref. 27 for individual underwater vehicles, and even in some swarm control works[8-15,17-19,30]; 4) *Control strategy and optimization algorithm*. Different from the multi-disciplinary optimization in ref. 27, this article develops the IRL-based swarm controller and optimization algorithm, which can reduce the dependence of the model.

The proposed DT-driven swarm platform can also be applied to other application scenarios, such as marine monitoring, off-shore exploration, submarine rescue, cooperation navigation and pipeline patrol. Nevertheless, there are still some unsolved issues for the swarm control for marine exploring. For example, underwater missions are performed by individual platforms without effective cross-platform coordination, which can result in limited scope of operations and low efficiency. To address this limitation, it is necessary to design a collaborative detection and tracking system utilizing an integrated cross-domain network comprising unmanned aerial vehicles (UAVs), unmanned surface vehicles (USVs) and AUVs.

Methods

Construction of digital twin model

As shown in Fig. 9, we consider a swarm of AUVs system for marine exploring, which mainly consists of one leader AUV and $N-1$ ($N>1$) follower AUVs. Particularly, we utilize the inertial reference frame (IRF) and body-fixed reference frame (BRF) to describe the cross-domain detection system. Without loss of generality, the DT models parameters are derived from the mapping of the above physical entities. Define $\Xi_{\rm AUV}$ as the parameter sets of AUV. Hence, the corresponding DT models parameter sets can be obtained, i.e., $\Xi_{\rm AUV}^{\rm DT}=\hat{\Xi}_{\rm AUV}$. In IRF, let $\hat{\boldsymbol{\eta}}_i=[\hat{\mathbf{p}}_i;\hat{\psi}_i]$ denote the position and orientation vector of $\Xi_{\rm AUV}^{\rm DT}$ $i\in\{1,...,N\}$, where $\hat{\mathbf{p}}_i=[\hat{x}_i,\hat{y}_i,\hat{z}_i]^{\rm T}$ and $\hat{\psi}_i$ is the yaw angle. Meanwhile, $\hat{x}_i$, $\hat{y}_i$ and $\hat{z}_i$ are the coordinates in the direction of the sway, surge, and heave, respectively. In BRF, let $\hat{\mathbf{v}}_i=[\hat{\mathbf{v}}_i^p;\hat{r}_i]$ denote the velocity vector of $\Xi_{\rm AUV}^{\rm DT}$ i, where $\hat{\mathbf{v}}_i^p=[\hat{u}_i,\hat{v}_i,\hat{w}_i]^{\rm T}$, $\hat{u}_i$, $\hat{v}_i$ and $\hat{w}_i$ are the linear velocities on surge, sway and heave, respectively. $\hat{r}_i$ is the yaw angle velocity. For the ocean currents disturbance in $\Xi_{\rm AUV}^{\rm DT}$, let $\hat{\mathbf{V}}_i^c=[\hat{\mathbf{v}}_i^c;0]$ and $\hat{\mathbf{v}}_i^c=[\hat{v}_{i,x}^c;\hat{v}_{i,y}^c;\hat{v}_{i,z}^c]$ denote the current velocity. $\hat{\mathbf{v}}_i^r=\hat{\mathbf{v}}_i-[\hat{\mathbf{v}}_i^J;0]$ is the relative velocity between the current and the $\Xi_{\rm AUV}^{\rm DT}$ i, where $\hat{\mathbf{v}}_i^J=\bar{\mathbf{J}}_i(\hat{\psi}_i)\hat{\mathbf{v}}_i^c$ and the rotation matrix $\bar{\mathbf{J}}_i(\hat{\psi}_i)=[\cos(\hat{\psi}_i),-\sin(\hat{\psi}_i),0;\sin(\hat{\psi}_i),\cos(\hat{\psi}_i),0;0,0,1]$. Then, the dynamic model of $\Xi_{\rm AUV}^{\rm DT}$ i can be described by

$$\begin{aligned}&\hat{\boldsymbol{\eta}}_i=\hat{\mathbf{J}}_i(\hat{\psi}_i)\hat{\mathbf{v}}_i^r+\hat{\mathbf{V}}_i^c,\\&\hat{\mathbf{M}}_i\dot{\hat{v}}_i+\hat{\mathbf{C}}_i^R\left(\hat{\mathbf{v}}_i\right)\hat{\mathbf{v}}_i+\hat{\mathbf{C}}_i^H\left(\hat{\mathbf{v}}_i^r\right)\hat{\mathbf{v}}_i^r+\hat{\mathbf{D}}_i\left(\hat{\mathbf{v}}_i^r\right)\hat{\mathbf{v}}_i^r=\hat{\boldsymbol{\tau}}_i,\end{aligned}\tag{1}$$

where $\hat{\mathbf{J}}_i(\hat{\psi}_i)$ =blkdiag$\{\bar{\mathbf{J}}_i(\hat{\psi}_i),1\}\in\mathcal{R}^{4\times4}$ is the rotation matrix of $\Xi_{\rm AUV}^{\rm DT}$ i, $\hat{\mathbf{M}}_i\in\mathcal{R}^{4\times4}$ is the inertia matrix, $\hat{\mathbf{D}}_i\left(\hat{\mathbf{v}}_i^r\right)\in\mathcal{R}^{4\times4}$ is the damping matrix, and $\hat{\boldsymbol{\tau}}_i=[\hat{\tau}_{u,i},\hat{\tau}_{v,i},\hat{\tau}_{w,i},\hat{\tau}_{r,i}]^{\rm T}$ is the control input vector. $\hat{\mathbf{C}}_i^R\left(\hat{\mathbf{v}}_{\rm i}\right)\in\mathcal{R}^{4\times4}$ and $\hat{\mathbf{C}}_i^H\left(\hat{\mathbf{v}}_i^r\right)\in\mathcal{R}^{4\times4}$ are the rigid body and hydrodynamics Coriolis-centripetal matrices, respectively. Some notations and their definitions are presented in Table 2.

For a clearer representation, we employ graph theory to characterize the topological structure of swarm system. The topology relationship of the system can be constructed by an *undirected graph* $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,...,N\}$ is the set of AUVs and $\mathcal{E}=\{(i,j):i,j\in\mathcal{V},j\neq i\}$ is the set of communication links.

Table 2: Notations and their definitions in this article.

Notation	Definition	Notation	Definition
p_c	Flow field position	p_t	Destination
v^c	Flow field velocity	γ_{ij}	Exprected relative distance
α^*	Optimal flow field parameter	e_c	Prediction flow field error
O_i	Obstacle position	e_{ij}	Swarm error
e_i	Matching error	$\hat{M}_i$	Inertia matrix
η_i	Position and orientation	$\hat{D}_i\left(\hat{v}_i^r\right)$	Damping matrix
v_i	Linear and yaw angel velocity	$\hat{C}_i^R\left(\hat{v}_\mathrm{i}\right)$	Rigid body matrix
$\hat{v}_i^r$	Relative velocity	$\hat{C}_i^H\left(\hat{v}_i^r\right)$	Hydrodynamics matrix
$\bar{J}_i(\hat{\psi}_i)$	Rotation matrix	M	Time windows
$\widehat{\alpha}$	Coefficients of flow filed	C_t	Total networking energy cost
B_m	$\overline{m}$-th obstacle information	$\overline{\psi}_i$	Estimated yaw velocity
p_{io}	Geometric center of obstacle	$\overline{v_i}$	Estimated linear velocity
ρ_{io}	Safe avoidance distance	$\overline{p}_i$	Estimated position
Θ_i	Set of detected obstacles	S_i^m	Number of reletive motion errors
$\overline{\alpha}$	Swarm error coefficient	$\mathrm{W}_{\tau_{i,k}}$	k-th column of W_{τ_i}
$k_{o,i}$	Weight for collision-free of AUVs	e_{pi}	Approximation error
k_{ij}	Obstacle avoidance weight	$\mathrm{t}_{i,\overline{s}}$	$\overline{s}$-th timestamp for collecting error
R	Positive definite weight matrix	$\widehat{W}_{\tau_i}$	Actor weight vector

Furthermore, the neighbor set of AUV $i \in \mathcal{V}$ is defined by $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$. Of note, to ensure topological connectivity and control stability, we assume that AUV i can obtain state information about neighboring AUVs via communication interactions, i.e., the undirected graph $\mathcal{G}$ is always connected.

Referring to ref. 36, we design the kernel function $\phi(\hat{\mathbf{p}}_i, \hat{\alpha})$ to model the 3D flow field based on Gaussian RBF, where $\hat{\mathbf{p}}_i$ and $\hat{\alpha}$ are the coefficient vectors of the current actual position of $\Xi_{\mathrm{AUV}}^{\mathrm{DT}}$ i and the unknown flow field, respectively. Specifically, the DT models of 3D flow field is modeled as follows:

$$\hat{\mathbf{v}}_i^c = \left[\begin{array}{ccc} \frac{\partial\phi(\mathbf{p}_i,\hat{\alpha})}{2\partial y\partial z}, & -\frac{\partial\phi(\mathbf{p}_i,\hat{\alpha})}{\partial x\partial z}, & \frac{\partial\phi(\mathbf{p}_i,\hat{\alpha})}{2\partial x\partial y} \end{array} \right]^{\mathrm{T}}, \tag{2}$$

where $\phi(\hat{\mathbf{p}}_i, \hat{\alpha}) = \sum_{p=1}^{P} \hat{\varpi}_p \exp(-\left\|\hat{\mathbf{p}}_i - \hat{\mathbf{c}}_p\right\|^2 / 2(\hat{\sigma}_p)^2)$, $p \in \{1, 2, ..., P\}$, P is the number of neurons, $\hat{\mathbf{c}}_p = [\hat{c}_p^x, \hat{c}_p^y, \hat{c}_p^z]^{\mathrm{T}}$ is the center vector of the p-th neuron, $\hat{\alpha}_p = [\hat{\varpi}_p; \hat{\mathbf{c}}_p; \hat{\sigma}_p]$ is the coefficients vector of the p-th neuron function, and $\hat{\alpha} = [\hat{\alpha}_1; ...; \hat{\alpha}_p; ...; \hat{\alpha}_P] \in \mathcal{R}^{5P\times 1}$ is the coefficients vector of the flow field.

Definition 1 (Obstacle Set): The underwater obstacles can be covered by convex spheres, and define the $\bar{m}$-th ($\bar{m} = 1, \ldots$) obstacle detected by AUV i as $\mathbf{B}_m(\mathbf{p}_{io}^{\bar{m}}, \rho_{io}^{\bar{m}})$, where $\mathbf{p}_{io}^{\bar{m}} = [x_{io}^{\bar{m}}, y_{io}^{\bar{m}}, z_{io}^{\bar{m}}]$ is the geometric center of the obstacle and $\rho_{io}^{\bar{m}} (\rho_{io}^{\bar{m}} > 0)$ denotes the safe avoidance distance. Based on this, the set of detected obstacles by AUV $i \in \{1, ..., N\}$ is defined as $\mathbf{\Theta}_i \subset \{\mathbf{B}_1(\mathbf{p}_{io}^1, \rho_{io}^1), \mathbf{B}_2(\mathbf{p}_{io}^2, \rho_{io}^2), \ \ldots\}$.

Of note, we consider the leader-follower swarm control of AUVs model, where one AUV is defined as the leader and the remaining AUVs are followers. Then, the destination position vector of AUVs is $\mathbf{p}_\mathrm{t} = [x_\mathrm{t}, y_\mathrm{t}, z_\mathrm{t}]^{\mathrm{T}}$. Along with this, the DT-driven swarm of AUVs system has the corresponding properties.

Properties: (1) The DT models parameter sets should asymptotically converge to physical parameter sets, i.e., $\Xi_{\mathrm{AUV}}^{\mathrm{DT}} \to \Xi_{\mathrm{AUV}}$ as $t \to \infty$. Meanwhile, this means that the virtual-real matching errors should asymptotically converge to zero; (2) The leader AUV can obtain the desired destination $\mathbf{p}_\mathrm{t}$, while the follower AUVs cannot know it; (3) The position vector of leader AUV should asymptotically converge to the expected destination $\mathbf{p}_\mathrm{t}$; (4) The swarm error $\mathbf{e}_{ij} = \mathbf{p}_i - \mathbf{p}_j - \boldsymbol{\gamma}_{ij}$ for AUV i should converge to zero, where $\boldsymbol{\gamma}_{ij}$ is the desired

relative position vector between AUVs i and $j \in \mathcal{N}_i$; (5) Each AUV should avoid collision with other AUVs, as well as with the obstacles in environment.

In addition, the dynamic model parameters ($\hat{\mathbf{M}}_i$, $\hat{\mathbf{D}}_i$, $\hat{\mathbf{C}}_i^R$, and $\hat{\mathbf{C}}_i^H$) of AUVs are obtained from the prior knowledge and AUVs, as similar to ref. 30. The above prior knowledge is stored in the parameter set $\Xi_{\text{AUV}}^{\text{DT}}$, and can be updated by the matching optimization algorithm.

Prediction of 3D flow field parameters

Based on the DT models, we need to further conduct flow field prediction to enhance underwater situation awareness. Particularly, we first collect the historical motion trajectories of AUVs, and then construct a flow field parameter estimator based on Newton iteration. Finally, the 3D flow field prediction can be achieved.

During the movement of swarm for AUVs, the interaction timeline of between physical AUV i and j is divided into M time windows (see Fig. 10). For example, $\mathcal{T}_i^1 = [t_i^0, t_i^1], \ldots, \mathcal{T}_i^m = [t_i^0, t_i^m], \ldots, \mathcal{T}_i^M = [t_i^0, t_i^M]$, where t_i^0 represents the initial moment, and $t_i^1, \ldots, t_i^m, \ldots, t_i^M$ are the end moments of M measurement windows, respectively. At time t_i^0, $t_i^1, \ldots, t_i^M$, the actual position of AUV i is measured by sensors, as denoted by $\mathbf{p}_i(t_i^0)$, $\mathbf{p}_i(t_i^1), \ldots, \mathbf{p}_i(t_i^M)$. At the same time, AUV i broadcasts a request signal to neighbor AUV j, and receives the historical position information of AUV j as $\mathbf{p}_j(t_i^{m-1}), \ldots, \mathbf{p}_j(t_i^m)$.

According to equation (2), the 3D flow field can be rewritten as:

$$\mathbf{v}_i^c(\mathbf{p}_i) = \begin{bmatrix} \frac{1}{2}\sum_{p=1}^{P} \phi_p(\mathbf{p}_i, \alpha) \frac{(y-c_p^y)(z-c_p^z)}{(\sigma_p)^4} \\ -\sum_{p=1}^{P} \phi_p(\mathbf{p}_i, \alpha) \frac{(x-c_p^x)(z-c_p^z)}{(\sigma_p)^4} \\ \frac{1}{2}\sum_{p=1}^{P} \phi_p(\mathbf{p}_i, \alpha) \frac{(x-c_p^x)(y-c_p^y)}{(\sigma_p)^4} \end{bmatrix}^{\mathrm{T}}. \tag{3}$$

Based on (1) and (3), the absolute motion integral error $\mathbf{e}_i^A$ at time t_i^m can be denoted as

$$\mathbf{e}_i^A(t_i^m) = \int_{t_i^0}^{t_i^m} (\dot{\mathbf{p}}_i(t) - \dot{\bar{\mathbf{p}}}_i(t))\mathrm{d}t = \int_{t_i^0}^{t_i^m} \mathcal{P}(p_i(t), \alpha)\mathrm{d}t, \tag{4}$$

where $\mathcal{P}(p_i(t), \alpha) = \bar{\mathbf{J}}_i(\psi_i)\mathbf{v}_i^{rp} + \mathbf{v}_i^c(\mathbf{p}_i) - \bar{\mathbf{J}}_i(\bar{\psi}_i)\bar{\mathbf{v}}_i$, $\mathbf{v}_i^{rp} = \mathbf{v}_i^p - \bar{\mathbf{J}}_i(\psi_i)\mathbf{v}_i^c(\mathbf{p}_i)$. $\bar{\psi}_i$ and $\bar{\mathbf{v}}_i$ are the estimated yaw and linear velocity, respectively. $\bar{\mathbf{p}}_i$ is the estimated position when $\mathbf{v}_i^c(\mathbf{p}_i) = 0$ in (3).

At time t_i^m, the absolute motion integral error $\mathbf{e}_i^A$ of AUV i can be written in vector form as follows: $\mathbf{E}_i^A = [\mathbf{e}_i^A(t_i^1); \mathbf{e}_i^A(t_i^2); \ldots; \mathbf{e}_i^A(t_i^{m-1}); \mathbf{e}_i^A(t_i^m)]$.

Besides that, we subdivide the timestamp sequence between t_i^{m-1} and t_i^m as $t_{i,1}^m, \ldots, t_{i,s}^m, \ldots, t_{i,S_i^m}^m$, where $s \in \{1, \ldots, S_i^m\}$. S_i^m is the number of relative motion integral errors. Then, the relative motion integral error $\mathbf{e}_{ij}^R$ at time $t_{i,s}^m$ can be denoted as

$$\mathbf{e}_{ij}^R(t_{i,s}^m) = \int_{t_i^0}^{t_{i,s}^m} (\dot{\mathbf{p}}_{ij}(t) - \dot{\bar{\mathbf{p}}}_{ij}(t))\mathrm{d}t = \int_{t_i^0}^{t_{i,s}^m} (\mathcal{P}(p_i(t), \alpha) - \mathcal{P}(p_j(t), \alpha))\mathrm{d}t. \tag{5}$$

where the relative position between AUV i and j is represented as $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$. Correspondingly, the estimated relative position can be denoted by $\bar{\mathbf{p}}_{ij} = \bar{\mathbf{p}}_j - \bar{\mathbf{p}}_i$.

Based on (5), the relative motion integral error between physical AUVs i and j during $\mathcal{T}_i^m$ is expressed as $\mathbf{E}_{ij}^R = \left[\left[\mathbf{e}_{ij}^R(t_{i,1}^1), \ldots, \mathbf{e}_{ij}^R(t_{i,S_i^1}^1)\right]^{\mathrm{T}}_{\text{1st window}}; \ldots; \left[\mathbf{e}_{ij}^R(t_{i,1}^m), \ldots, \mathbf{e}_{ij}^R(t_{i,S_i^m}^m)\right]^{\mathrm{T}}_{m-\text{th window}} \right]$.

For the prediction of flow field, a nonlinear constraint equation can be constructed via the motion integral errors $\mathbf{E}_i^A$ and $\mathbf{E}_{ij}^R$. During the $\mathcal{T}_i^m$, the relative motion integral error of AUV i is $\mathbf{E}_{ij}^R = [\mathbf{E}_{i1}^R; \ldots; \mathbf{E}_{ij}^R; \ldots; \mathbf{E}_{i\mathcal{N}}^R]$,

where $j \in \{1, \ldots, \mathcal{N}\}$ represents the AUV j. According to (4) and (5), the nonlinear constraint equation can be denoted by

$$\underbrace{\begin{bmatrix} \mathbf{E}_{ij}^{R} \\ -- \\ \mathbf{E}_{i}^{A} \end{bmatrix}}_{\delta_i^E} = \underbrace{\begin{bmatrix} \int_{t_i^0}^{t_{i,1}^1} (\mathcal{P}(p_i(t),\alpha) - \mathcal{P}(p_1(t),\alpha))\mathrm{d}t \\ \vdots \\ \int_{t_i^0}^{t_{i,S_i^m}^m} (\mathcal{P}(p_i(t),\alpha) - \mathcal{P}(p_{\mathcal{N}}(t),\alpha))\mathrm{d}t \\ - - - - - - - - - - - - \\ \int_{t_i^0}^{t_i^m} \mathcal{P}(p_i(t),\alpha)\mathrm{d}t \end{bmatrix}}_{\mathcal{V}_i^E(\alpha) \in \mathcal{R}^{\bar{S}\times 1}}, \tag{6}$$

where $\bar{S} = 3\sum_{\bar{m}=1}^{m}(S_i^{\bar{m}}\mathcal{N} + \bar{m})$ is the number of rows of the vector mentioned above.

Next, we define α^* as the optimal iterative parameter of the flow field. Thereby, the optimization problem of flow field parameters can be expressed as $\alpha^* = \arg\min_{\alpha} \left\| \mathcal{V}_i^E(\alpha) - \delta_i^E \right\|$.

To solve α^*, a parameter iterative estimator is proposed. We first select an initial parameter value $\alpha^{(0)}$, and then design the update rate of velocity field parameters based on Newton iteration. The specific expression is as follows:

$$\alpha^{(s+1)} = \alpha^{(s)} + \lambda \left(\left(L(\alpha^{(s)}) \right)^{\mathrm{T}} L(\alpha^{(s)}) \right)^{-1} \left(L(\alpha^{(s)}) \right)^{\mathrm{T}} (\delta_i^E - \mathcal{V}_i^E(\alpha^{(s)})), \tag{7}$$

with

$$L(\alpha^{(s)}) = \frac{\partial \mathcal{V}_i^E(\alpha^{(s)})}{\partial \alpha^{(s)}} = [\frac{\partial \left(\mathcal{V}_i^E(\alpha^{(s)})\right)^1}{\partial \alpha^{(s)}}; \ldots; \frac{\partial \left(\mathcal{V}_i^E(\alpha^{(s)})\right)^{\bar{s}}}{\partial \alpha^{(s)}}; \ldots; \frac{\partial \left(\mathcal{V}_i^E(\alpha^{(s)})\right)^{\bar{S}}}{\partial \alpha^{(s)}}] \in \mathcal{R}^{3\bar{S}\times 5P},$$

$$\frac{\partial \left(\mathcal{V}_i^E(\alpha^{(s)})\right)^{\bar{s}}}{\partial \alpha^{(s)}} = \begin{cases} \int_{t_i^0}^{t_{i,\bar{s}}} [\frac{\partial \left(\mathbf{v}_i^c(\mathbf{p}_i) - \mathbf{v}_j^c(\mathbf{p}_j)\right)}{\partial \alpha_1^{(s)}} \quad \cdots \quad \frac{\partial \mathbf{v}_i^c(\mathbf{p}_i)}{\partial \alpha_P^{(s)}}] \mathrm{d}t, & \bar{s} \in \{1, \ldots, \bar{S}-1\}, \\ \int_{t_i^0}^{t_i^m} [\frac{\partial \mathbf{v}_i^c(\mathbf{p}_i)}{\partial \alpha_1^{(s)}}, \ldots, \frac{\partial \mathbf{v}_i^c(\mathbf{p}_i)}{\partial \alpha_P^{(s)}}] \mathrm{d}t, & \bar{s} = \bar{S}, \end{cases}$$

where λ is the step size for updating flow field parameters, $\left(\mathcal{V}_i^E(\alpha^{(s)})\right)^{\bar{s}}$ represents the $\bar{s}$-th row of data for $\mathcal{V}_i^E(\alpha^{(s)})$, $t_{i,\bar{s}}$ is the $\bar{s}$-th timestamp for collecting motion integral errors of AUV i, $\alpha^{(s+1)}$ and $\alpha^{(s)}$ are the $s+1-$th and s-th iterations, respectively.

Finally, the termination condition for (7) can be denoted by $\|\alpha^{(s+1)} - \alpha^{(s)}\| \leq \delta_\alpha$, where δ_α is the maximum tolerable estimated error. Of note, the value of δ_α is absolutely derived from system requirements, specifically by balancing performance needs with the consequences and cost of errors.

Energy-efficient swarm networking design

The communication topology of all AUVs can be denoted by an *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For AUV i, its degree is represented as d_i. Without loss of generality, we make $d_i < 3$ to reduce energy consumption of network. Let dis_{ij} denote the shortest distance between AUVs i and j for interaction. Thereby, the total energy cost can be defined as: $C_t = \sum mission_{ij} * dis_{ij}$, where $i, j \in \{1, 2, ..., N\}$ $(i \neq j)$, and $mission_{ij}$ is the size of data should be interfaced between AUVs i and j.

The energy minimization problem for the swarm of AUVs network can be represented as

$$\begin{aligned} \min_{\mathcal{E}} C_t \quad &= \quad \min_{\mathcal{E}} \sum mission_{ij} * dis_{ij} \\ &\text{s.t.} \begin{cases} \forall i \in \mathcal{V},\ d_i \leq 3 \\ \text{the topology graph is connected} \\ \forall u, v \in \mathcal{V}, \text{ if } \mathcal{E}_{uv} \notin \mathcal{E}, \text{ then } d_u = 3 \text{ or } d_v = 3 \end{cases} \end{aligned} \tag{8}$$

where $\mathcal{E}_{uv}$ is the communication edge between AUVs u and v. Of note, $d_i \leq 3$ is used to limit the number of network ports to control economic costs. The optimal solution is non-decreasing when edges are added subject to the degree constraint $d \leq 3$, whenever at least one endpoint of each new edge has degree 3, i.e., if $\mathcal{E}_{uv} \notin \mathcal{E}$, then $d_u = 3$ or $d_v = 3$.

In order to further reduce the communication load, we divide the time axis into two windows, as shown in Fig. **??**. Specifically, AUV $j \in \mathcal{N}_i$ broadcasts position information to AUV $i \in \{1, ..., N\}$ only in the first part of T_w. For the other parts of T_w, we use the position from the DT models to achieve the swarm task, where AUV j and i do not require interaction. Based on this, we can obtain the swarm error, i.e.,

$$\mathbf{e}_{ij} = \begin{cases} \mathbf{p}_i - \mathbf{p}_j - \boldsymbol{\gamma}_{ij}, & \text{for the first part of } T_w \\ \hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j - \boldsymbol{\gamma}_{ij}, & \text{for the other parts of } T_w \end{cases} \tag{9}$$

The merits of the above design can reduce the communication energy consumption, since AUV $i \in \{1, ..., N\}$ does not need to exchange information with its neighbors at each time instant. Meanwhile, the accumulation errors can be avoided by periodically updating $\mathbf{e}_{ij}$ with the data exchange.

Swarm controller design for AUVs

According to (9), the swarm error vector $\mathbf{e}_{ij}$ between AUVs $i \in \{1, ..., N\}$ and $j \in \mathcal{N}_i$ can be obtained. Meanwhile, the tracking error from leader AUV 1 to the desired destination is denoted by $\mathbf{e}_{1\text{tar}} = \mathbf{p}_i - \mathbf{p}_\text{t}$. Then, the cost function for collision-free swarm control can be denoted as

$$g_i(\mathbf{p}_i, \boldsymbol{\tau}_i) = \sum_{j \in \mathcal{N}_i} (\bar{\alpha} \left\| \mathbf{e}_{ij} \right\|^2 + \frac{k_{ij}}{\left\| \|\mathbf{p}_i - \mathbf{p}_j\| - \mathbf{R}_\# \right\|^2}) + \beta \left\| \mathbf{e}_{1\text{tar}} \right\|^2 + \sum_{m \in \Theta_i} \frac{k_{o,i}}{\left\| \|\mathbf{p}_i - \mathbf{p}_{io}^m\| - \rho_{io}^m \right\|^2} + \boldsymbol{\tau}_i^\text{T} \mathbf{R} \boldsymbol{\tau}_i, \tag{10}$$

where $\bar{\alpha}$ is the swarm error coefficient and $\mathbf{R}_\#$ is the minimum safe distance for any two AUVs. If $i = 1$, $\beta = 1$, otherwise $\beta = 0$. In addition, $k_{o,i}$ penalizes collisions between AUVs, where a larger value increases repulsive force at close range; k_{ij} penalizes proximity to obstacles, preventing approaching behaviors; and $\mathbf{R}$ is a positive definite matrix that weights the control torque, balancing performance and energy consumption.

During the $[t, t+T]$ time period, the value function can be defined as

$$V_i(\mathbf{p}_i(t)) = \int_t^\infty g_i(\mathbf{p}_i, \boldsymbol{\tau}_i) \mathrm{d}t = \int_t^{t+T} g_i(\mathbf{p}_i, \boldsymbol{\tau}_i) \mathrm{d}t + V_i(\mathbf{p}_i(t+T)). \tag{11}$$

Then, we employ IRL to acquire the optimal control policy $\boldsymbol{\tau}_i^*$, such that the optimal value function $V_i^*(\mathbf{p}_i)$ satisfies $V_i^*(\mathbf{p}_i) = \min_{\tau_i} V_i(\mathbf{p}_i)$. Particularly, we denote V_i^n as the updated value function to reduce the dependence of model parameters, and $\boldsymbol{\tau}_i^n$ is regarded as the updated control input in the n-th iteration. Differentiating value function (11) with respect to t, we can have

$$\dot{V}_i^n = -g_i(\mathbf{p}_i, \boldsymbol{\tau}_i^n) - 2(\boldsymbol{\tau}_i^{n+1})^\text{T} \mathbf{R} (\boldsymbol{\tau}_i - \boldsymbol{\tau}_i^n). \tag{12}$$

Integrating both sides of (12), one has

$$V_i^n(\mathbf{p}_i(t+T)) - V_i^n(\mathbf{p}_i(t)) = -\int_t^{t+T} (g_i(\mathbf{p}_i, \boldsymbol{\tau}_i^n) + 2(\boldsymbol{\tau}_i^{n+1})^\text{T} \mathbf{R} (\boldsymbol{\tau}_i - \boldsymbol{\tau}_i^n)) \mathrm{d}t. \tag{13}$$

Next, the policy iteration is conducted to obtain $\boldsymbol{\tau}_i^*$. The specific steps are as follows:

1) *Initialization:* Implement an initialized control policy $\boldsymbol{\tau}_i^0$, and collect the position information $\mathbf{p}_i$.

2) *Policy Evaluation:* Solve value function at iteration step n : $V_i^n(\mathbf{p}_i(t)) = \int_t^{t+T} g_i(\mathbf{p}_i, \boldsymbol{\tau}_i^n) \mathrm{d}t + V_i^n(\mathbf{p}_i(t+T))$.

3) *Policy Improvement:* Update the control policy as $\boldsymbol{\tau}_i^{n+1} = -\mathbf{R}^{-1} \bar{\mathbf{B}}_i^\text{T} (\partial V_i^n / \partial \mathbf{X}_i)/2$.

Utilizing the basis functions to approximate V_i^n and $\boldsymbol{\tau}_i^{n+1}$, the estimations of V_i^n and $\boldsymbol{\tau}_i^{n+1}$ are denoted as

$$\hat{V}_i^n(\mathbf{p}_i)=\hat{\mathbf{W}}_{v_i}^n\boldsymbol{\varpi}_{v_i}(\mathbf{p}_i),\ \hat{\boldsymbol{\tau}}_i^{n+1}(\mathbf{p}_i)=\hat{\mathbf{W}}_{\tau_i}^{n+1}\boldsymbol{\varpi}_{\tau_i}(\mathbf{p}_i), \tag{14}$$

where $\boldsymbol{\varpi}_{v_i}(\mathbf{p}_i)$ and $\boldsymbol{\varpi}_{\tau_i}(\mathbf{p}_i)$ are basis functions. $\hat{\mathbf{W}}_{v_i}$ and $\hat{\mathbf{W}}_{\tau_i}$ are weight vectors.

Let $\mathbf{v}^1=[v_1^1,\ v_2^1,\ v_3^1,\ v_4^1]^{\mathrm{T}}=\boldsymbol{\tau}_i-\boldsymbol{\tau}_i^n$ and $\mathbf{R}=\mathrm{diag}\{r_{1,i},\ r_{2,i},\ r_{3,i},\ r_{4,i}\}$. Besides, $\hat{W}_{\tau_i,k}$ is the k-th column of $\hat{\mathbf{W}}_{\tau_i}$. Based on (13) and (14), one has

$$\mathbf{e}_{pi}=\hat{\mathbf{W}}_{v_i}^n\boldsymbol{\varepsilon}_{v_i}+\textstyle\sum_{k=1}^4\hat{W}_{\tau_i,k}^{n+1}\varepsilon_{u_i,k}+\int_t^{t+T}g_i(\mathbf{p}_i,\boldsymbol{\tau}_i^n)\mathrm{d}t, \tag{15}$$

where $\boldsymbol{\varepsilon}_{v_i}=\boldsymbol{\varpi}_{v_i}(\mathbf{p}_i(t+T))-\boldsymbol{\varpi}_{v_i}(\mathbf{p}_i(t))$ and $\varepsilon_{u_i,k}=2r_{k,i}\int_t^{t+T}\boldsymbol{\varpi}_{\tau_i}(\mathbf{X}_i(t))v_k^1\mathrm{d}t$ $(k=1,2,3,4)$.

The approximation error $\mathbf{e}_{pi}$ is represented as

$$y_{pi}+\mathbf{e}_{pi}(t)=\hat{\mathbf{W}}_{pi}\mathbf{x}_{pi}, \tag{16}$$

where $y_{pi}=-\int_t^{t+T}g_i(\mathbf{P}_i,\boldsymbol{\tau}_i^n)\mathrm{d}\tau$, $\hat{\mathbf{W}}_{pi}=[\hat{W}_{v_i}^n,\ \hat{W}_{\tau_i,1}^{n+1},\ \hat{W}_{\tau_i,2}^{n+1},\ \hat{W}_{\tau_i,3}^{n+1},\ \hat{W}_{\tau_i,4}^{n+1}]$ and $\mathbf{x}_{pi}=[\boldsymbol{\varepsilon}_{v_i}^{\mathrm{T}},\ \varepsilon_{u_i,1}^{\mathrm{T}},\ \varepsilon_{u_i,2}^{\mathrm{T}},\ \varepsilon_{u_i,3}^{\mathrm{T}},\ \varepsilon_{u_i,4}^{\mathrm{T}}]^{\mathrm{T}}$. $\hat{W}_{v_i}$ and $\hat{W}_{\tau_i}$ are the critic and actor weights, respectively.

To minimize the error $\mathbf{e}_{pi}(t)$, the least-squares method can be used to update the weights. Along with this, the error function can be denoted by

$$J=\left\|\hat{\mathbf{W}}_{pi}\mathbf{X}_{pi}-\mathbf{Y}_{pi}\right\|^2=\left\|\mathbf{X}_{pi}^{\mathrm{T}}\hat{\mathbf{W}}_{pi}^{\mathrm{T}}-\mathbf{Y}_{pi}\right\|^2=(\mathbf{X}_{pi}^{\mathrm{T}}\hat{\mathbf{W}}_{pi}^{\mathrm{T}}-\mathbf{Y}_{pi})^{\mathrm{T}}(\mathbf{X}_{pi}^{\mathrm{T}}\hat{\mathbf{W}}_{pi}^{\mathrm{T}}-\mathbf{Y}_{pi}). \tag{17}$$

Let the derivative of (17) equal to zero, i.e., $\partial J/\partial\hat{\mathbf{W}}_{pi}^{\mathrm{T}}=2\mathbf{X}_{pi}\mathbf{X}_{pi}^{\mathrm{T}}\hat{\mathbf{W}}_{pi}-2\mathbf{X}_{pi}\mathbf{Y}_{pi}=0$. Then, we obtain that $\hat{\mathbf{W}}_{pi}$ can be updated as $\hat{\mathbf{W}}_{pi}=(\mathbf{X}_{pi}\mathbf{X}_{pi}^T)^{-1}\mathbf{X}_{pi}\mathbf{Y}_{pi}$, where $\mathbf{X}_{pi}=[x_{pi,1},\ldots,x_{pi,k}]$ and $\mathbf{Y}_{pi}=[y_{pi,1},\ldots,y_{pi,k}]^{\mathrm{T}}$.

When $\left\|V_i^n-V_i^{n-1}\right\|\leq\delta_V$ and $\left\|\boldsymbol{\tau}_i^{n+1}-\boldsymbol{\tau}_i^n\right\|\leq\delta_\tau$, the above weight strategy iteration terminates, such that the optimal value function and control policy can be obtained, where δ_V and δ_τ are small positive thresholds. To sum up, we repeat the above procedures to obtain the final optimal control policy $\boldsymbol{\tau}_i^*$ and value function $V_i^*(\mathbf{p}_i)$. Similarly, the swarm controller of the DT models is also designed in this way to obtain the optimal control policy $\hat{\boldsymbol{\tau}}_i^*$. For the readability, the design procedure is omitted here.

Virtual-real matching error optimizing algorithm

With the swarm controller, the AUVs and DT models update their own position information. However, DT models still face challenges in achieving high-precision reproduction of physical entities due to practical limitations: sensor errors, manual inaccuracies, unknown variables, and so on. Thereby, the error optimizing algorithm is designed periodically at the first part of T_{w} to reduce the matching errors between AUVs and DT models.

Based on (1), we find the fact that the model parameter $\Xi_{\mathrm{AUV}}^{\mathrm{DT}}$ are related to the position information $\hat{\mathbf{p}}_i$. Then, the matching error $\mathbf{e}_i$ can be defined, i.e., $\mathbf{e}_i=\hat{\mathbf{p}}_i-\mathbf{p}_i$.

To make $\Xi_{\mathrm{AUV}}^{\mathrm{DT}}\rightarrow\Xi_{\mathrm{AUV}}$, the cost function for $\mathbf{e}_i$ is

$$g_{ei}(\mathbf{e_i},\boldsymbol{\phi_i})=\mu\left\|\mathbf{e}_i\right\|^2+\boldsymbol{\phi}_i^{\mathrm{T}}\mathbf{Q}\boldsymbol{\phi}_i, \tag{18}$$

where μ is an error coefficient, $\mathbf{Q}$ is a positive definite matrix, and $\boldsymbol{\phi}_i$ is a vector composed of $\Xi_{\mathrm{AUV}\ i}^{\mathrm{DT}}$.

The value function for virtual-real error $\mathbf{e}_i$ can be obtained as

$$V_{ei}(\mathbf{e_i}(t))=\int_t^{t+T}g_{ei}(\mathbf{e_i},\boldsymbol{\phi_i})\mathrm{d}t+V_{ei}(\mathbf{e_i}(t+T), \tag{19}$$

and the optimal characteristic parameter vector $\boldsymbol{\phi}_i^*$ is

$$\boldsymbol{\phi}_i^* = \text{argmin}_{\boldsymbol{\phi}_i}\{\int_t^{t+T} g_{ei}(\mathbf{e_i}, \boldsymbol{\phi_i})\mathrm{d}t + V_{ei}(\mathbf{e_i}(t+T)\}. \tag{20}$$

The value function is to minimize the virtual-real error $\mathbf{e}_i$, such that the characteristic parameter vector $\boldsymbol{\phi}_i$ can be optimized. Of note, the final convergence conditions are represented as $\left\|V_{ei}^n - V_{ei}^{n-1}\right\| \leq \delta_{V_{ei}}$ and $\left\|\boldsymbol{\phi}_i^{n+1} - \boldsymbol{\phi}_i^n\right\| \leq \delta_{\boldsymbol{\phi}_i}$, where $\delta_{V_{ei}}$ and $\delta_{\boldsymbol{\phi}_i}$ are the maximum tolerable convergence errors. Ultimately, the value function can maximize the accumulation reward of the DT models to obtain the optimal strategy, i.e., $\Xi_{\text{AUV}}^{\text{DT}} \to \Xi_{\text{AUV}}$ as $t \to \infty$. The optimization procedure is similar to the one in swarm controller, and hence omitted here.

Data availability

The data used in this work are available from the corresponding authors upon request. Of note, the videos of the experiment are provided by https://v.youku.com/v_show/id_XNjQ4NDQwOTY3Mg==.html or https://youtu.be/k_zwlnsfhvU.

Code availability

The part code of the DT-driven swarm control design used in this study is publicly available at https://docs.google.com/document/d/1nySs8usqpjfMXz4i0U2ds3yeS6VQt6F5NIy1z-guPUI/edit?usp=sharing

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62222314 and the Hebei Provincial Department of Education Graduate Innovation Ability Training Funding Project CXZZBS2025065.

Author contributions

J.Y. and T.Y.Z. and wrote the article. X.P.G. conceived the project. T.Y.Z. performed the experiments. J.Y., X.Y., and C.L.C. analyzed the data, and all authors provided feedback.

Competing interests

The authors declare no competing interests.

References

1 Brotherson, M. & Jr, S.W. A message from island leaders: protect the Pacific Ocean from deep-sea mining. *Nature* **642**, 299-302 (2025).

2 Jaffe, J. et al. A swarm of autonomous miniature underwater robot drifters for exploring submesoscale ocean dynamics. *Nat. Commun.* **8**, 14189 (2017).

3 Li, G., Wong, TW., Shih, B. et al. Bioinspired soft robots for deep-sea exploration. *Nat. Commun.* **14**, 7097 (2023).

4 Cohen, N.R. et al. Microeukaryote metabolism across the western North Atlantic Ocean revealed through autonomous underwater profiling. *Nat. Commun.* **15**, 7325 (2024).

5 Sharma, S. et al. Low power, non-intrusive 3D localization for underwater mobile robots. *Commun. Eng.* **4**, 93 (2025).

6 Zhang, Y. et al. Multi-AUV cooperative control and autonomous obstacle avoidance study. *Ocean Eng.* **304**, 117634 (2024).

7 Hu, X., Cao, Z. & Chen, Y. Dynamic task assignment and path optimization for multi-AUVs system. *IEEE Trans. Intell. Transp. Syst.* **26**, 8233-8246 (2025).

8 Li, G. et al. Bioinspired soft robots for deep-sea exploration. *Nat. Commun.* **14**, 7097 (2023).

9 Zhao. Q. et al. Bio-inspired swarm of underwater robots: a review. *Bioinspir. Biomim.* **20,** 1-28 **(2025).**

10 Zhen, Q. et al. Formation control of a multi-AUVs system based on virtual structure and artificial potential field on SE(3). *Ocean Eng.* **253**, 111148 (2022).

11 Pang, W., Zhu. D. & Sun. C. Multi-AUV formation reconfiguration obstacle avoidance algorithm based on affine transformation and improved artificial potential field under ocean currents disturbance. *IEEE Trans. Autom. Sci. Eng.* **21**, 1469-1487 (2024).

12 Zhang, Y. et al. 3D hybrid formation control of an underwater robot swarm: Switching topologies, unmeasurable velocities, and system constraints. *ISA Trans.* **136,** 345-360, (2023)**.**

13 Yan, Z. et al. Limited communication consensus control of leader-following multi-UUVs in a swarm system under multi-independent switching topologies and time delay. *IEEE Access* **6**, 33183-33200 (2018).

14 Alamdari, S. et al. Cooperative impedance control for multiple underwater vehicle manipulator systems under lean communication. *IEEE J. Ocean. Eng.* **46**, 447-465 (2021).

15 Wang, L. et al. A novel obstacle avoidance consensus control for multi-AUV formation system. *IEEE/CAA J. Autom. Sinica* **10,** 1304-1318 (2023).

16 Li, Z., Chitre, M. & Stojanovic, M. Underwater acoustic communications. *Nat. Rev. Electr. Eng.* **2**, 83-95 (2025).

17 Liu, Y. et al. Computational fluid dynamics prediction of the dynamic behavior of autonomous underwater vehicles. *IEEE J. Ocean. Eng.* **45**, 724-739 (2020).

18 Yin, J. & Wang, N. Predictive trajectory tracking control of autonomous underwater vehicles based on variable fuzzy predicto. *Int. J. Fuzzy Syst.* **23**, 1809–1822 (2020).

19 Yan, J. et al. Integrated localization and tracking for AUV with model uncertainties via scalable sampling-based reinforcement learning approach. *IEEE Trans. Syst. Man Cybern. Syst.* **52**, 6952–6967 (2022).

20 Zheng, Y. et al. High-turbidity underwater image enhancement via turbidity suppression fusion. *IEEE Trans. Circuits Syst. Video Technol.* **35**, 3527-3540 (2025).

21 Bokhtiar Al Zami, M. et al. Digital twin in industries: A comprehensive survey. *IEEE Access* **13**, 47291-47336 (2025).

22 Tao, F. & Qi, Q. Make more digital twins. *Nature* **573**, 490-491 (2019).

23 Xia, P. et al. ROV teleoperation via human body motion mapping: Design and experiment. *Comput. Ind.* **150**, 103959 (2023).

24 Bartolucci, V. et al. A digital twin infrastructure for designing an underwater survey with a professional DPV. in *Proc. 30th Mediterranean Conf. Control Autom.* pp. 829-834 (2022).

25 Gao, L. et al. A digital twin-driven trajectory tracking control method of a lower-limb exoskeleton. *Control Eng. Pract.* **127**, 105271 (2022).

26 Yang, J. et al. A digital twins enabled underwater intelligent internet vehicle path planning system via reinforcement learning and edge computing. *Digit. Commun. Netw.* **10**, 282-291 (2024).

27 Yang, M. et al. Digital twin-driven industrialization development of underwater gliders. *IEEE Trans. Ind. Inform.* **19**, 9680-9690 (2023).

28 Yang, X. et al. Social learning with actor–critic for dynamic grasping of underwater robots via digital twins. *Ocean Eng.* vol. **306**, 118070 (2024).

29 Park, S. et al. Enhancement of control performance for degraded robot manipulators using digital twin and proximal policy optimization. *IEEE Access*, **12**, 19569-19583 (2024).

30 Zhang, T. et al. Digital twin-driven formation control of ROVs: An integral reinforcement learning-based solution. *IEEE Trans. Ind. Inform.* **20**, 14277-14286 (2024).

31 Zhou, J. et al. Consensus estimation of flow field via underwater sensor networks in inhomogeneous medium. *IEEE Trans. Circuits II.* **72**, 633-637 (2025).

32 Lu, J., Yang, X. & Wang, J. Velocity vector estimation of two-dimensional flow field based on STIV. *Sensors* **23**, 955 (2023).

33 Chen, H. et al. Decentralized estimation of ocean current field using underwater acoustic sensor networks. *J. Acoust. Soc. Am.* **149**, 3106-3121 (2021).

34 Mealey, R.M. A method for calculating error probabilities in a radar communication system. *IEEE Trans. Space Electron. Telemetry* **9**, 3742 (1963).

35 Serra, M. et al. Search and rescue at sea aided by hidden flow structures. *Nat. Commun.* **11**, 2525 (2020).

36 He, Y. et al. A tree-based distributed method for cooperative flow field estimation. *Syst. Control Lett.* **175**, 105511 (2023).

Figure Legends

Fig. 1 **Description of the swarm control of autonomous underwater vehicles (AUVs), where AUVs transmit data to buoys through acoustic communication. After that, buoys utilize electromagnetic communication to relay data to the control center via the assistance of satellites.**

Fig. 2 **Effectiveness of digital twin (DT)-driven swarm controller in the pool. (a)** Experiment setup in the pool. **(b)** Experiment setup in the Unity. **(c)** Real flow field. **(d)** Predicted flow field. **(e)** Prediction error. **(f)** Trajectories of autonomous underwater vehicles (AUVs). **(g)** Trajectories of the DT models. **(h)** Norm of the swarm errors. **(i)** Relative distances with obstacles. **(j)** Relative distances among AUVs. **(k)** Norm of the matching errors.

Fig. 3 **Overall structural relationship diagram.**

Fig.4 **Comparison with other swarm results in the pool. (a)** Trajectories without digital twin (DT) models. **(b)** Communication energy. **(c)** Trajectories of autonomous underwater vehicles (AUVs) without the matching algorithm. **(d)** Trajectories of DT models without the matching algorithm. **(e)** Trajectories of AUVs with the matching algorithm in this article. **(f)** Trajectories of DT models with the matching algorithm in this article.

Fig.5 **Effectiveness of digital twin (DT)-driven swarm controller in the lake. (a)** Experiment setup in the lake. **(b)** Experiment setup in the Unity. **(c)** Real flow field. **(d)** Predicted flow field. **(e)** Prediction error. **(f)** Trajectories of autonomous underwater vehicles (AUVs). **(g)** Trajectories of the DT models. **(h)** Norm of the swarm errors. **(i)** Relative distances with obstacles. **(j)** Relative distances among AUVs. **(k)** Norm of the matching errors.

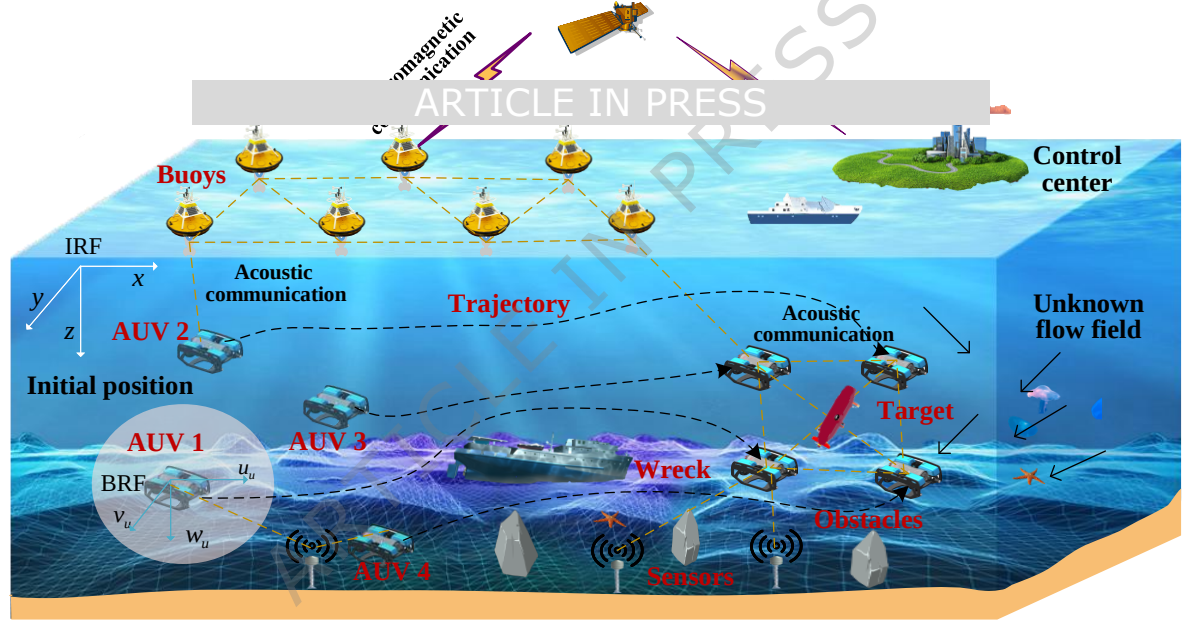
Fig.6 **Comparison with other swarm results in the lake. (a)** Trajectories without digital twin (DT) models. **(b)** Communication energy. **(c)** Trajectories of autonomous underwater vehicles (AUVs) without the matching algorithm. **(d)** Trajectories of DT models without the matching algorithm. **(e)** Trajectories of AUVs with the matching algorithm in this article. **(f)** Trajectories of DT models with the matching algorithm in this article.

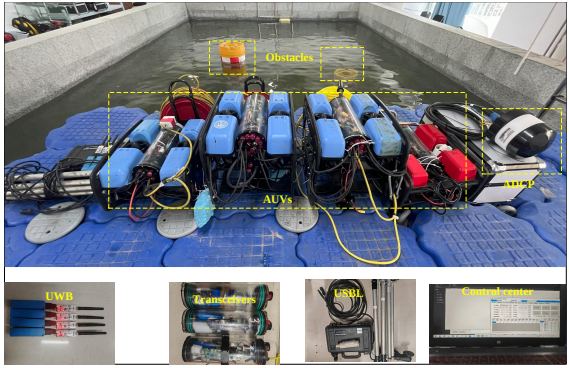
Fig.7 **Effectiveness of digital twin (DT)-driven swarm controller in the sea. (a)** Experiment setup in the sea. **(b)** Experiment setup in the Unity. **(c)** Real flow field. **(d)** Predicted flow field. **(e)** Prediction error. **(f)** Trajectories of autonomous underwater vehicles (AUVs). **(g)** Trajectories of the DT models. **(h)** Norm of the swarm errors. **(i)** Relative distances with obstacles. **(j)** Relative distances among AUVs. **(k)** Norm of the matching errors.

Fig.8 **Comparison with other swarm results in the near-shore of sea. (a)** Trajectories without digital twin (DT) models. **(b)** Communication energy. **(c)** Trajectories of autonomous underwater vehicles (AUVs) without the matching algorithm. **(d)** Trajectories of DT models without the matching algorithm. **(e)** Trajectories of AUVs with the matching algorithm in this article. **(f)** Trajectories of DT models with the matching algorithm in this article.

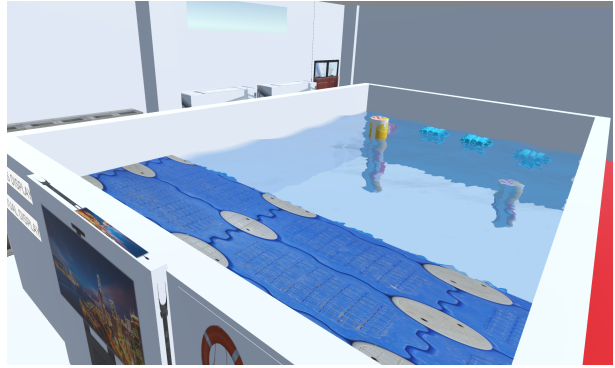
Fig.9 **Overall architecture of digital twin-driven swarm control of autonomous underwater vehicles.**

Fig.10 **Description of the m-th measurement window.**

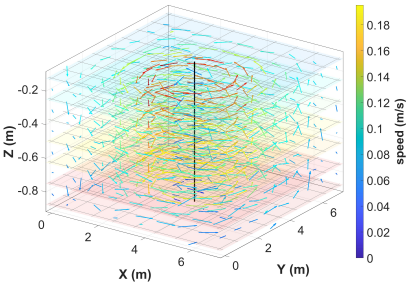




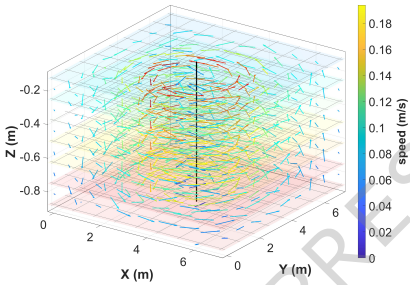
(a)



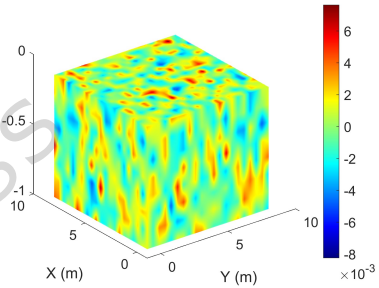
(b)



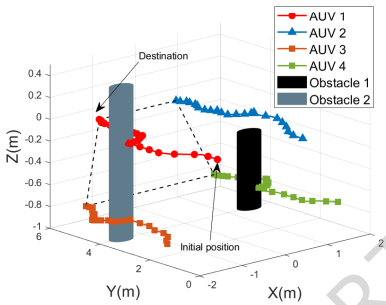
(c)



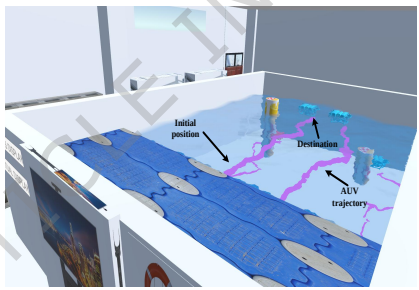
(d)



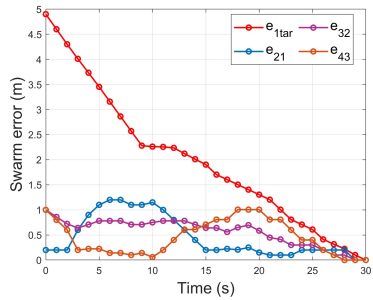
(e)



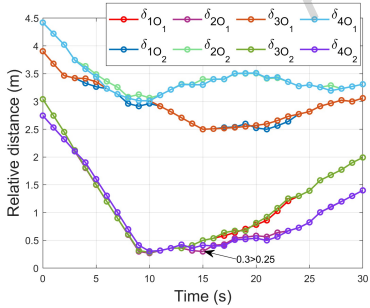
(f)



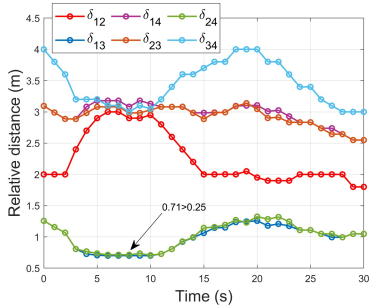
(g)



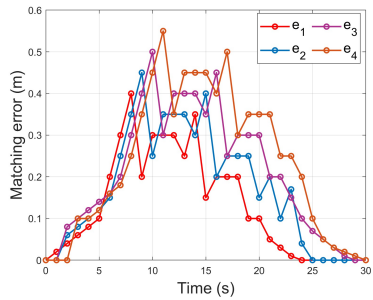
(h)



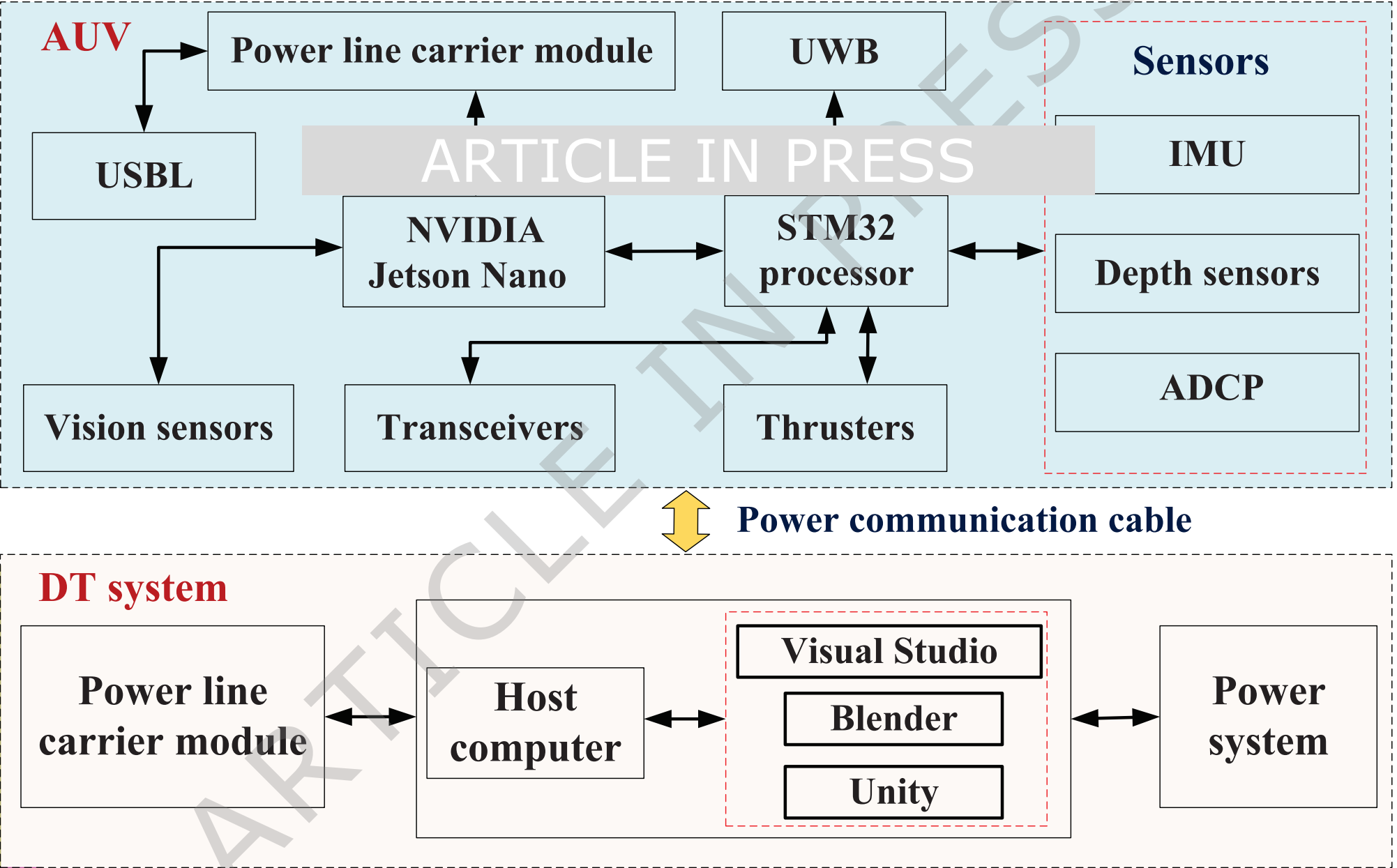
(i)

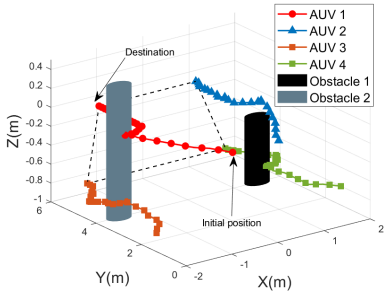


(j)

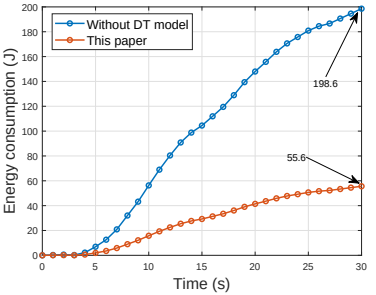


(k)

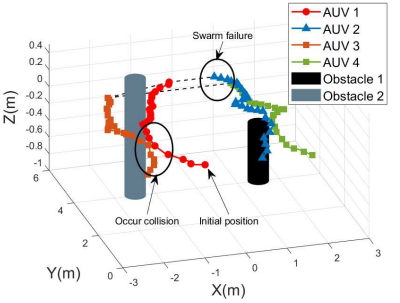



AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Destination
Initial position
Z(m)
Y(m)
X(m)

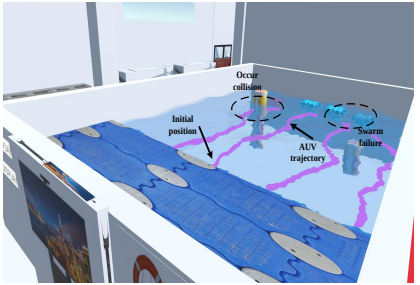
(a)


Without DT model
This paper
198.6
55.6
Energy consumption (J)
Time (s)

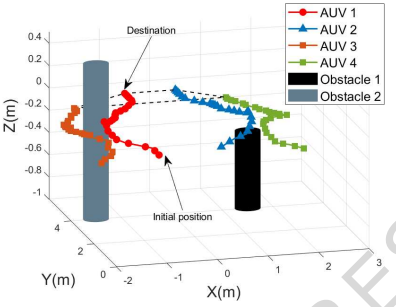
(b)


AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Swarm failure
Occur collision
Initial position
Z(m)
Y(m)
X(m)

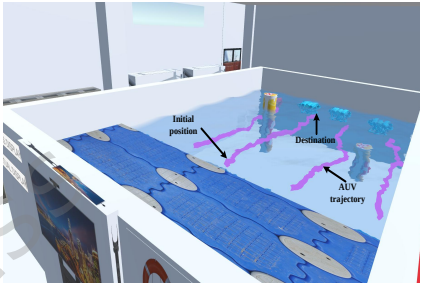
(c)


Occur collision
Initial position
Swarm failure
AUV trajectory

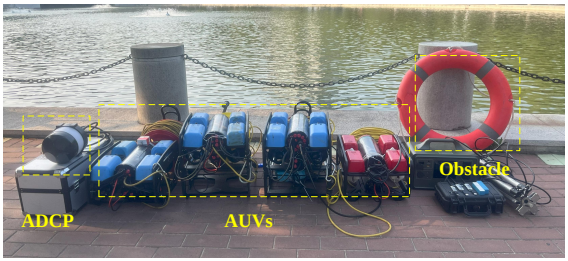
(d)

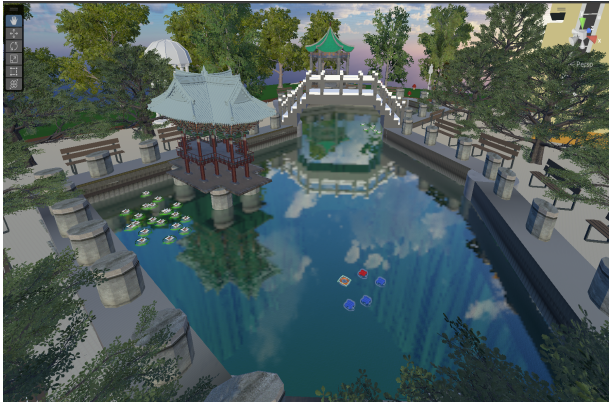

AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Destination
Initial position
Z(m)
Y(m)
X(m)

(e)

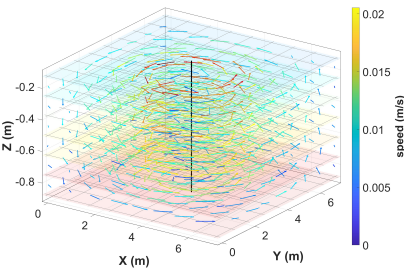

Initial position
Destination
AUV trajectory

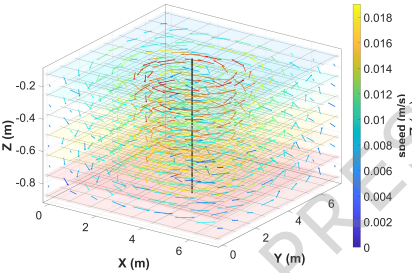
(f)

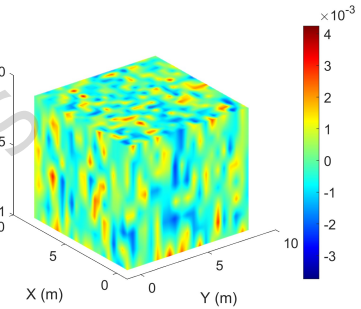




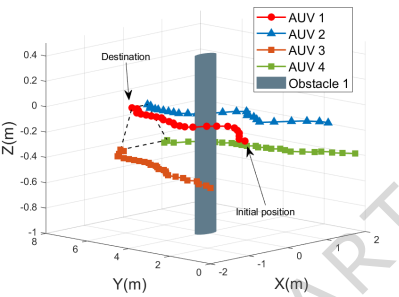
(a) (b)

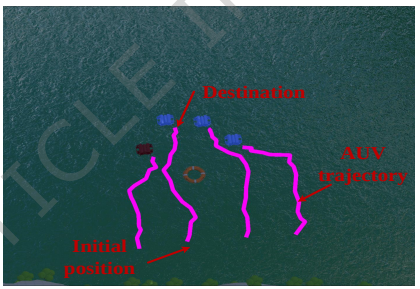


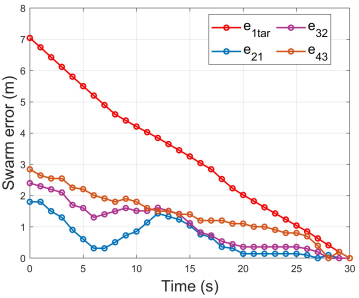




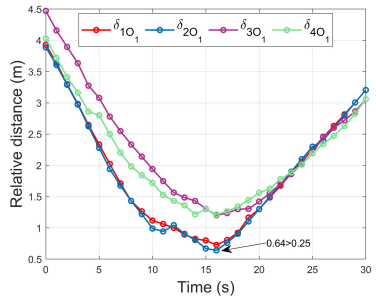
(c) (d) (e)

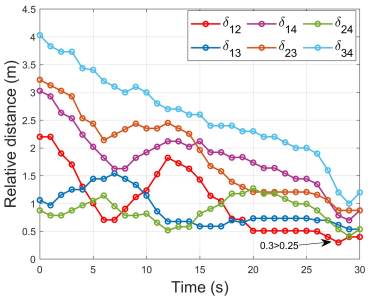


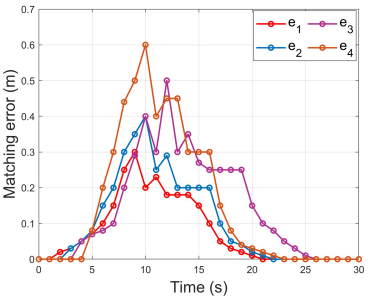




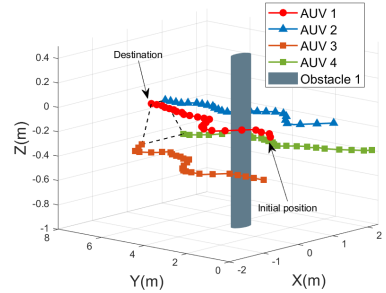
(f) (g) (h)



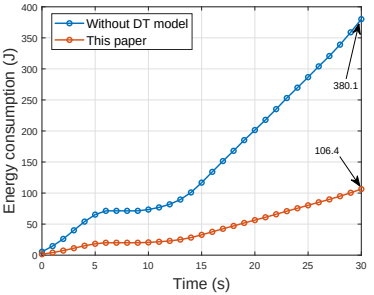




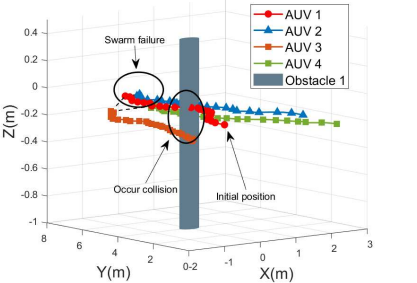
(i) (j) (k)


AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Destination
Initial position
Z(m)
Y(m)
X(m)

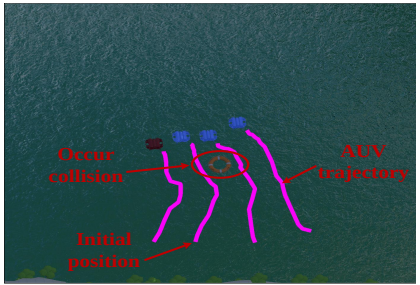
(a)


Without DT model
This paper
380.1
106.4
Energy consumption (J)
Time (s)

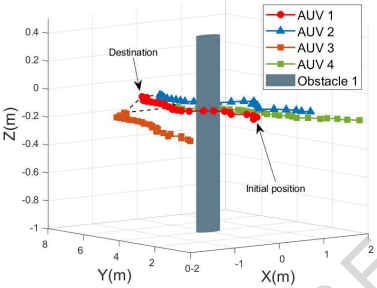
(b)


AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Swarm failure
Occur collision
Initial position
Z(m)
Y(m)
X(m)

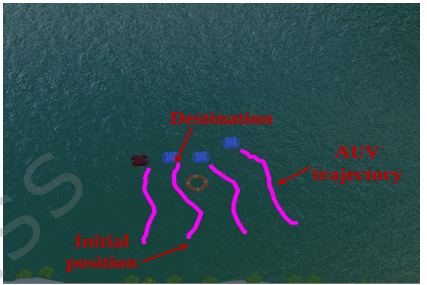
(c)


Occur collision
AUV trajectory
Initial position

(d)

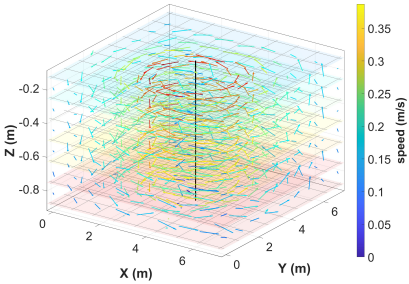

AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Destination
Initial position
Z(m)
Y(m)
X(m)

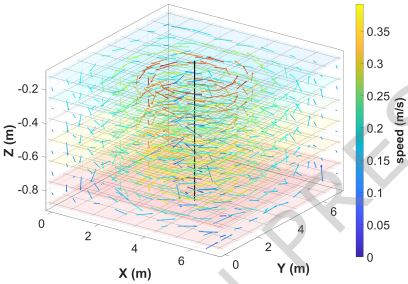
(e)

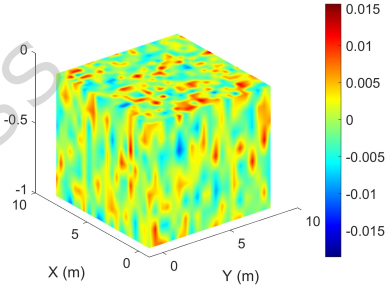

Destination
AUV trajectory
Initial position

(f)

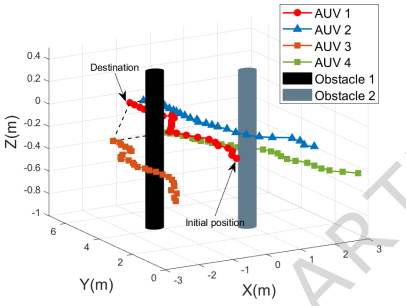
(a) (b)

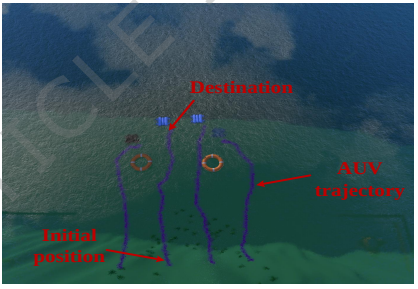


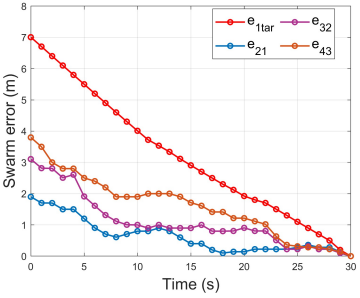




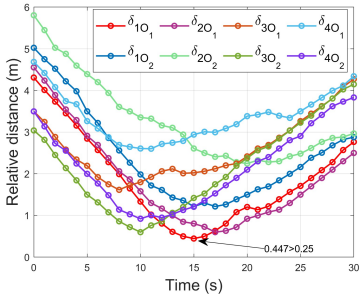
(c) (d) (e)

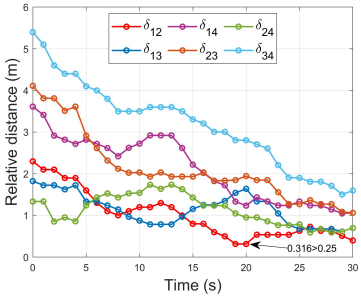


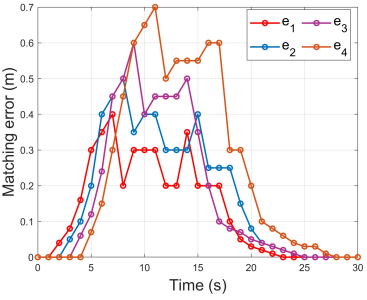




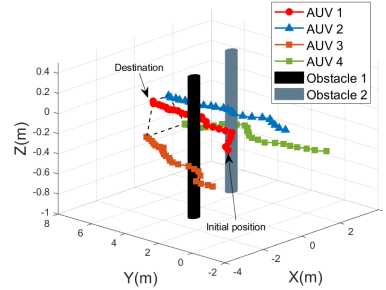
(f) (g) (h)



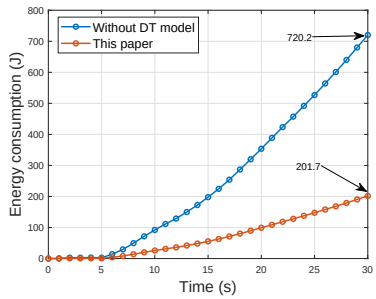




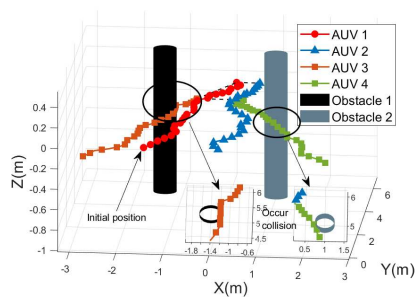
(i) (j) (k)


AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Destination
Initial position
Z(m)
Y(m)
X(m)

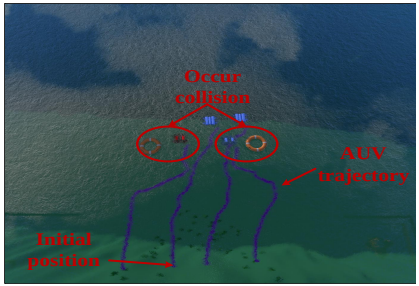
(a)


Without DT model
This paper
720.2
201.7
Energy consumption (J)
Time (s)

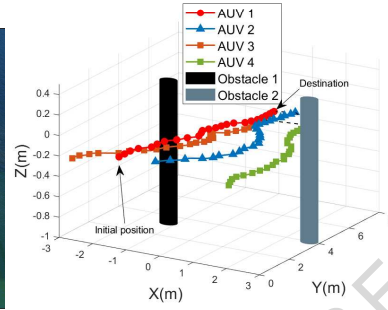
(b)


AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Initial position
Occur collision
Z(m)
X(m)
Y(m)

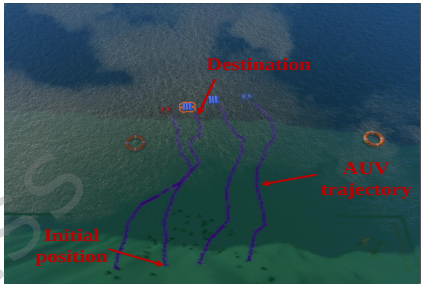
(c)


Occur collision
AUV trajectory
Initial position

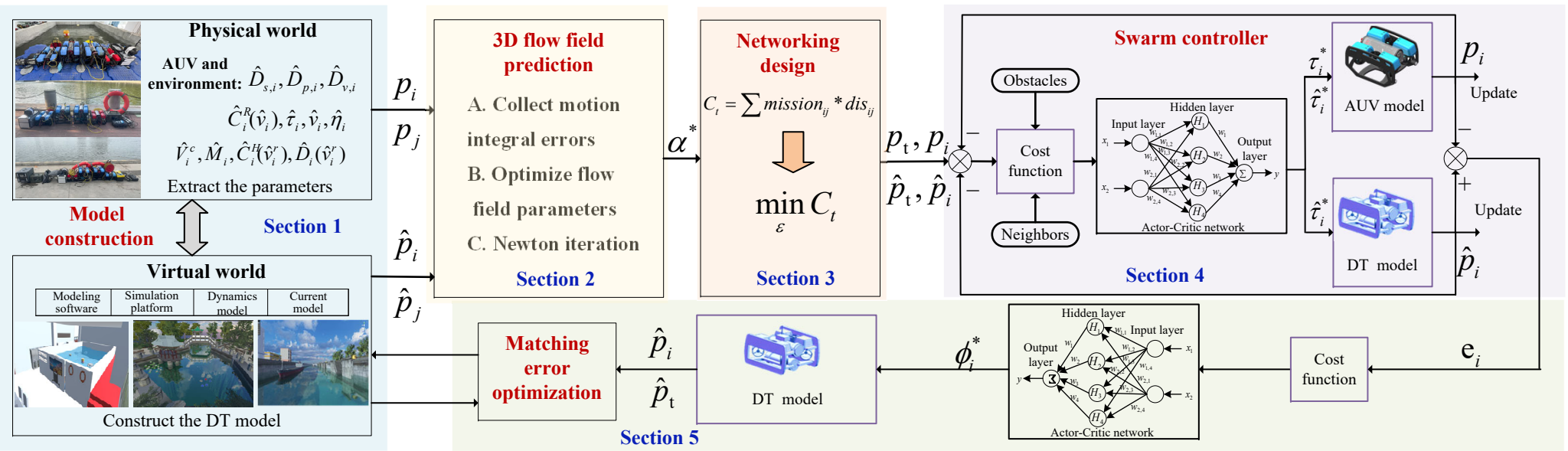
(d)

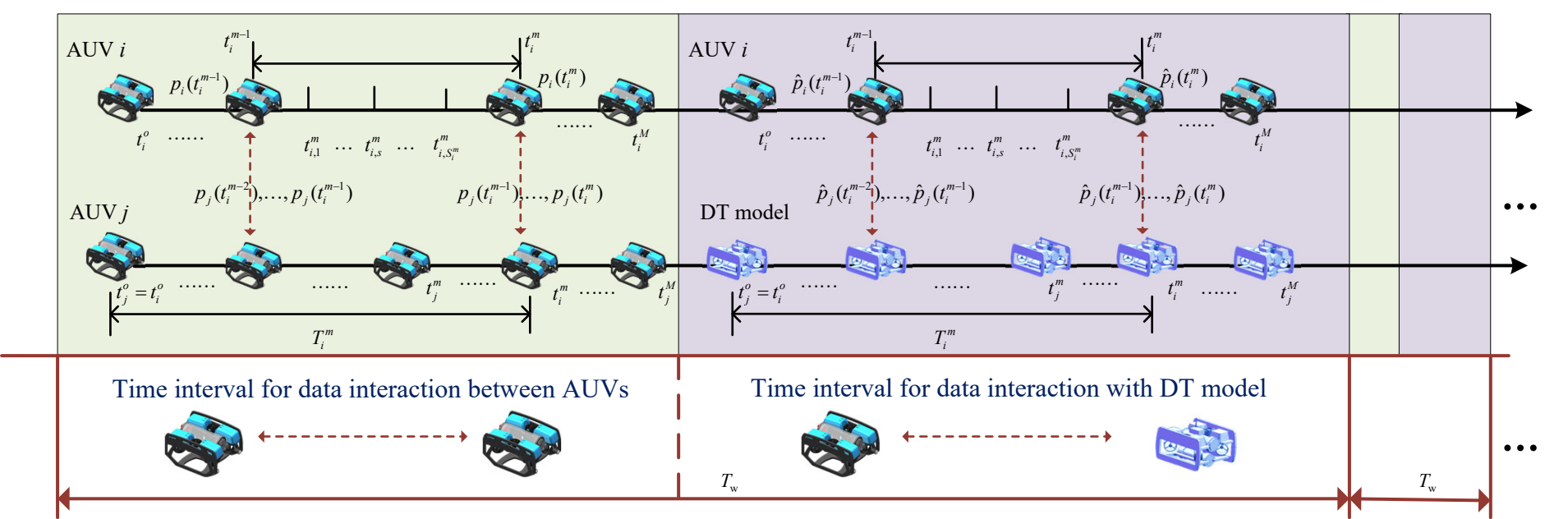

AUV 1
AUV 2
AUV 3
AUV 4
Obstacle 1
Obstacle 2
Destination
Initial position
Z(m)
X(m)
Y(m)

(e)


Destination
AUV trajectory
Initial position

(f)


Physical world
AUV and environment: $\hat{D}_{s,i}, \hat{D}_{p,i}, \hat{D}_{v,i}$
$\hat{C}_i^R(\hat{v}_i), \hat{\tau}_i, \hat{v}_i, \hat{\eta}_i$
$\hat{V}_i^c, \hat{M}_i, \hat{C}_i^H(\hat{v}_i^r), \hat{D}_i(\hat{v}_i^r)$
Extract the parameters
Model construction
Section 1
Virtual world
Modeling software
Simulation platform
Dynamics model
Current model
Construct the DT model
p_i
p_j
$\hat{p}_i$
$\hat{p}_j$
3D flow field prediction
A. Collect motion integral errors
B. Optimize flow field parameters
C. Newton iteration
Section 2
α^*
Networking design
$C_t = \sum mission_{ij} * dis_{ij}$
$\min_{\varepsilon} C_t$
Section 3
p_t, p_i
$\hat{p}_t, \hat{p}_i$
Swarm controller
Obstacles
Neighbors
Cost function
Hidden layer
Input layer
Output layer
Actor-Critic network
τ_i^*
$\hat{\tau}_i^*$
AUV model
DT model
p_i
Update
Update
$\hat{p}_i$
Section 4
Matching error optimization
$\hat{p}_i$
$\hat{p}_t$
DT model
Section 5
ϕ_i^*
Hidden layer
Input layer
Output layer
Actor-Critic network
Cost function
e_i


AUV i
t_i^{m-1}
t_i^m
$p_i(t_i^{m-1})$
$p_i(t_i^m)$
t_i^o
$t_{i,1}^m$ … $t_{i,s}^m$ … $t_{i,S_i^m}^m$
t_i^M
AUV j
$p_j(t_i^{m-2}),\ldots,p_j(t_i^{m-1})$
$p_j(t_i^{m-1}),\ldots,p_j(t_i^m)$
$t_j^o = t_i^o$
t_j^m
t_i^m
t_j^M
T_i^m
AUV i
$\hat{p}_i(t_i^{m-1})$
$\hat{p}_i(t_i^m)$
t_i^o
$t_{i,1}^m$ … $t_{i,s}^m$ … $t_{i,S_i^m}^m$
t_i^M
DT model
$\hat{p}_j(t_i^{m-2}),\ldots,\hat{p}_j(t_i^{m-1})$
$\hat{p}_j(t_i^{m-1}),\ldots,\hat{p}_j(t_i^m)$
$t_j^o = t_i^o$
t_j^m
t_i^m
t_j^M
T_i^m
Time interval for data interaction between AUVs
Time interval for data interaction with DT model
T_w
T_w