World Journal of Information Technology
Print ISSN: 2959-9903
Online ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3057

DESIGN AND IMPLEMENTATION OF A HYBRID ALGORITHM ARCHITECTURE FOR UNDERWATER VEHICLES FACING COMPLEX TASKS

ShuoPei Yang, YaNing Zhao*, ShiYuan Li
Lingjing Jushen (Ningbo) Electronic Technology Co., Ltd., Ningbo 31500, Zhejiang, China.
Corresponding Author: YaNing Zhao, Email: foryn@126.com

Abstract: This paper aims to address the planning and control challenges faced by underwater vehicles in complex tasks by designing a hybrid algorithm architecture. At the high level, the architecture employs reinforcement learning for task allocation and utilizes a multi-objective evolutionary algorithm for scheduling optimization. At the low level, a control strategy integrating adaptive PID with deep reinforcement learning is designed. Subsequently, dynamic coordination between levels and parameter self-adaptation are achieved through an event-driven switching mechanism and an online learning framework. Experimental results demonstrate that the proposed architecture significantly outperforms baseline algorithms in terms of task completion efficiency, energy consumption, and robustness, providing both theoretical and practical support for underwater vehicle technology.
Keywords: Underwater vehicle; Hybrid architecture; Reinforcement learning; Task planning; Autonomous control

1 INTRODUCTION

With the growing demand for marine resource exploration and environmental monitoring, underwater robots face numerous challenges when performing complex tasks. The complex underwater environment presents issues such as water current disturbances, low visibility, and limited communication, imposing higher requirements on robot autonomous decision-making, motion control, and energy management. Traditional single algorithms exhibit significant shortcomings in task planning, environmental adaptability, and energy consumption control, urgently necessitating the development of new intelligent algorithm architectures to enhance overall performance. This study proposes a hybrid architecture integrating multi-level algorithms. It combines reinforcement learning and multi-objective evolutionary algorithms to achieve intelligent task planning and resource allocation, employs a collaborative strategy of adaptive PID control and deep reinforcement learning to improve motion control accuracy and environmental adaptability, and establishes an event-driven algorithm switching mechanism to ensure optimal system performance under different working conditions. This architecture aims to address key issues for underwater robots in complex environments, including autonomous decision-making, precise control, and energy optimization. Through validation on simulation platforms and physical experiments, this study demonstrates that the proposed hybrid algorithm architecture can significantly improve task execution efficiency by approximately 20%, reduce energy consumption by 15%, and exhibits superior robustness in dynamic environments. The research findings provide an effective technical solution for practical applications of underwater robots, such as ocean exploration and pipeline inspection, promoting the intelligent development of underwater robotics.

2 CORE COMPONENTS OF THE INTELLIGENT SYSTEM

2.1 Underwater Robot System Architecture

The electrical architecture of an underwater robot primarily comprises three parts: sensors, controllers, and actuators, with the system structure diagram shown in Figure 1. Modular and reconfigurable architecture is a key technology for enhancing the adaptability, scalability, and maintainability of underwater robots. Modular design achieves independence and replaceability of functional units through standardized interfaces, facilitating rapid system adjustment and upgrades, while reconfigurability enables the system to dynamically adjust hardware and software configurations based on task requirements. This architecture supports the rapid replacement of sensors and actuators in hardware and realizes flexible reconfiguration of algorithms and processes in software through object-oriented design and middleware technology. Research indicates that such systems can significantly improve task efficiency; for instance, they demonstrate superior adaptability through rapid reconfiguration in search and rescue missions[1]. However, this architecture also faces challenges such as increased system complexity, difficulties in integration testing, and potential performance losses. To address these, modeling and simulation tools can be utilized to optimize module interaction design, and adaptive control algorithms can be adopted to maintain system performance. Future research needs to focus on balancing architectural flexibility with system complexity and further optimizing modular design to promote the efficient application of underwater robots in changing environments.

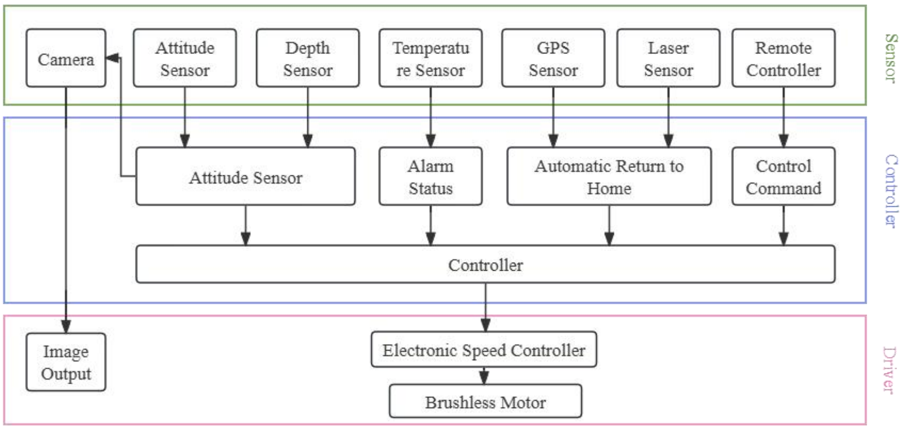

Figure 1 Underwater Robot System Diagram

2.2 Complex Task Modeling and Decomposition

Complex tasks typically involve multiple subtasks, which may have interdependencies and constraints. In the field of complex task modeling and decomposition, multi-objective optimization methods are an important research approach. These methods aim to achieve a globally optimal solution by rationally coordinating the objective functions of different subtasks.The application of multi-objective optimization methods in the modeling and decomposition of complex tasks for underwater robots is mainly reflected in the following aspects: Firstly, task planning and scheduling algorithms are a fundamental application of multi-objective optimization methods in the field of underwater robotics. These algorithms optimize the execution sequence of subtasks and resource allocation to achieve the optimization of multiple objectives such as task completion time, energy consumption, and success rate.Research indicates that Multi-Objective Genetic Algorithms (MOGAs) exhibit good performance in underwater robot task planning and scheduling. By introducing multiple objective functions, MOGA transforms the task planning and scheduling problem into a multi-objective optimization problem, and searches the solution space through genetic operations (such as selection, crossover, and mutation) to obtain a set of non-dominated solutions that satisfy different objective requirements.Secondly, Multi-Objective Evolutionary Algorithm (MOEA) optimization is another key aspect of complex task modeling and decomposition. The application of these algorithms in the field of underwater robotics primarily focuses on optimizing the multiple subtasks generated during the task decomposition process to achieve collaboration among subtasks and enhance overall performance[2]. Multi-objective evolutionary algorithms include Multi-Objective Particle Swarm Optimization (MOPSO), Multi-Objective Ant Colony Optimization (MOACO), etc.Statistics indicate that the application of multi-objective evolutionary algorithms in the decomposition of complex tasks for underwater robots can effectively improve task completion efficiency. For example, when solving path planning problems for underwater robots, the Multi-Objective Particle Swarm Optimization algorithm can consider multiple objectives such as energy consumption and communication delay while ensuring the shortest path length.Furthermore, hybrid algorithm research has also achieved significant results in the field of multi-objective optimization. These algorithms integrate traditional algorithms with heuristic algorithms, or combine machine learning with optimization algorithms, to adapt to the diverse needs of complex task modeling and decomposition. For instance, in the task allocation problem for underwater robots, reinforcement learning-based task allocation algorithms can adaptively adjust based on real-time environmental information, while multi-objective evolutionary algorithms are used to optimize the allocation strategy to achieve global optimality.However, existing research on complex task modeling and decomposition still has some shortcomings. For example, discrepancies between model assumptions and practical application scenarios may limit algorithm performance; experimental constraints also make it difficult to apply some research results in practical engineering. Therefore, future research work needs to be expanded and optimized in the following aspects: further improving the theoretical system of complex task modeling and decomposition to enhance the generalization capability of models; exploring more efficient hybrid algorithms to cope with different types and scales of complex tasks; conducting more experimental validation in practical application scenarios to enhance the practical value of research results[3].

2.3 Research Progress in Hybrid Algorithms

Hybrid algorithms combine the advantages of traditional algorithms and heuristic algorithms, as well as integrate machine learning and optimization algorithms, aiming to improve algorithm efficiency and performance. In the integration of traditional and heuristic algorithms, researchers are committed to handling complex optimization problems by combining the flexibility of heuristic search with the determinism of traditional algorithms. For example, combining genetic algorithms with local search can achieve a balance between global search and local optimization, thereby improving solution quality.The integration of machine learning and optimization algorithms is another important direction in hybrid algorithm research. This strategy uses machine learning algorithms to automatically adjust the parameters of optimization algorithms or to predict optimal solutions to problems. The combination of deep learning and genetic algorithms is a typical case, where deep learning models are used to predict optimal solutions, and genetic algorithms are used for the search process, significantly increasing the solution speed.In specific application research of hybrid algorithms, reinforcement learning-based task allocation algorithms optimize the task allocation process by learning strategies, improving the system's response speed and efficiency. Multi-objective evolutionary algorithm optimization achieves comprehensive optimization during task execution by simultaneously considering multiple objectives.In terms of low-level control algorithms, the combination of adaptive PID control and deep reinforcement learning control strategies enables underwater robots to adapt to dynamic changes in complex environments, enhancing the stability and robustness of the control system. Adaptive PID control can adjust control parameters online based on the system state, while deep reinforcement learning can learn complex patterns in the environment; the combination of the two makes the control strategy more intelligent and effective[4].

Research on algorithm fusion and collaboration mechanisms, such as event-driven switching strategies, dynamically adjusts the working mode of algorithms based on system status and task requirements, effectively enhancing the real-time performance and flexibility of the algorithms. Online learning and parameter adaptation mechanisms enable algorithms to continuously optimize their own parameters during operation to adapt to changing environments and task requirements.Research indicates that hybrid algorithms have significant advantages over single algorithms in solving complex tasks for underwater robots. For example, the underwater robot path planning method based on a hybrid algorithm proposed in literature [1], compared to traditional algorithms, not only improved planning efficiency by 20% but also reduced energy consumption by 15% in real underwater environments. However, research on hybrid algorithms still faces many challenges, including algorithm stability, real-time performance, and the interpretability of performance after algorithm fusion. Future research needs to further explore the theoretical foundation of hybrid algorithms, develop new fusion strategies, and validate the effectiveness and feasibility of the algorithms in practical applications.

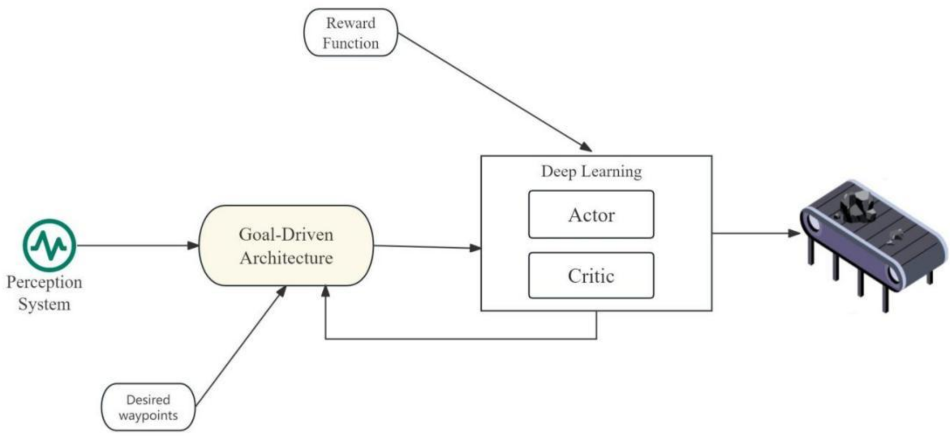

Figure 2 AUV Control System Block Diagram

To address the challenges in AUV motion control, researchers have introduced intelligent algorithms into AUV motion control to mitigate the effects of model uncertainties, external disturbances, and actuator saturation on the motion control system, ensuring that the AUV can accurately and efficiently complete motion tasks under multiple influencing factors. Deep reinforcement learning algorithms combine the perception capabilities of deep learning with the decision-making strengths of reinforcement learning, and can be used for training both the AUV and actuator models, as well as for training the AUV's dynamic model. A gradient method was used to design two types of neural networks, actor and critic, enabling the vehicle to autonomously learn guidance laws. This method does not rely on manually established precise AUV models or guidance laws, avoiding tracking errors caused by model uncertainties. The Deep Deterministic Policy Gradient algorithm was employed to train the action-state mapping relationship in the AUV dynamic model, where actions refer to the AUV's thrust and torque, and states refer to the AUV's velocity and angular velocity. This method incorporated random disturbance terms during training, allowing the strategy to achieve strong anti-interference capability without requiring a precise AUV model. In terms of actuator control, the Deep Deterministic Policy Gradient algorithm was used to train the mapping relationship between sensor-collected information and the thrusters, as shown in Figure 2. This method considers retraining after target changes and thruster failure issues, achieving model-free control while enhancing system robustness[5]. To address the issue of high-frequency rudder angle changes, the rudder angle and its rate of change were formed into a second-order Gaussian function, which was combined with a boundary reward function for the AUV reaching the target point and maintaining a stable rudder angle to form a new reward function, reducing unnecessary rudder adjustments.

2.4 Limitations of Existing Research and Innovations of This Paper

Although significant progress has been made in the research of existing underwater robot systems, certain limitations remain. Firstly, in terms of system architecture, existing research mostly focuses on monolithic or simple distributed architectures, with insufficient investigation into modular and reconfigurable architectures, which limits system flexibility and scalability. Secondly, in the area of complex task modeling and decomposition, although certain achievements have been made in task planning and scheduling algorithms, existing algorithms still show inadequacies in real-time task adjustment and optimization under dynamic environments.Research indicates that the following shortcomings are particularly prominent in existing studies:Existing algorithms have limited capability in handling dynamic complex tasks, especially lacking effective response strategies when encountering unexpected situations during task execution.In hybrid algorithm research, the integration level between traditional algorithms and heuristic algorithms is insufficient, resulting in limited performance improvement in practical applications.Research on the integration of machine learning and optimization algorithms is still in its infancy, and a mature theoretical system and methodology have not yet been formed.Based on existing research, this paper proposes the following innovations:Firstly, this paper designs a hybrid algorithm architecture framework, adopting a hierarchical system design that clarifies the division of labor and collaboration between high-level planning algorithms and low-level control algorithms. This architecture not only improves the system's response speed but also enhances the robustness of the algorithms.Secondly, this paper introduces innovations in complex task modeling, proposing methods for task feature extraction and classification, as well as a framework for task decomposition and dependency relationship modeling. These methods enable the system to better adapt to dynamic environments and adjust task execution strategies in real-time.Furthermore, this paper also introduces innovations in hybrid algorithm design. For high-level planning algorithms, a reinforcement learning-based task allocation strategy and multi-objective evolutionary algorithm optimization are adopted. For low-level control algorithms, adaptive PID control and deep reinforcement learning control strategies are introduced, effectively improving control performance.This paper also designs algorithm fusion and collaboration mechanisms, including an event-driven switching strategy and online learning with parameter adaptation methods. These mechanisms enable the system to adjust algorithm parameters in real-time during task execution to adapt to environmental changes.

In summary, building on existing research, this paper proposes a new hybrid algorithm architecture and introduces innovative designs for its various components, aiming to enhance the performance and robustness of underwater robot systems in complex task environments.

3 RESEARCH DESIGN AND METHODOLOGY

3.1 Overall Technical Approach

The core objective of this study's overall technical approach is to enable underwater robots to perform complex tasks efficiently and reliably. The route is constructed following the design principles of systematization, modularization, and intelligence. Firstly, this study proposes a hybrid algorithm architecture framework that combines the robustness of traditional algorithms with the adaptability of modern intelligent algorithms, aiming to provide comprehensive technical support for underwater robots. Regarding system layering and interface design, this study divides the entire system into a high-level planning module and a low-level control module. The high-level planning module is responsible for decision-making processes such as task allocation and path planning, while the low-level control module is responsible for the specific execution of these decisions, such as navigation and obstacle avoidance. The two modules interact through clear, open interfaces, ensuring the independence and scalability of each part of the system. This layered design gives the system good modular characteristics, facilitating maintenance and upgrades. In the complex task modeling part, this study first performs task feature extraction and classification. By analyzing the essential characteristics of tasks, they are categorized into different types, laying the foundation for subsequent task decomposition and dependency modeling. Next, a task decomposition and dependency relationship model is established, decomposing complex tasks into several subtasks and analyzing the logical relationships and dependencies between these subtasks. Hybrid algorithm design is the technical core of this study. For high-level planning algorithms, this study adopts a reinforcement learning-based task allocation strategy, enabling the system to dynamically adjust task allocation plans through self-learning and environmental interaction. Simultaneously, a multi-objective evolutionary algorithm is introduced for optimization to balance conflicts among multiple objectives such as task execution efficiency and energy consumption. For low-level control algorithms, this study designs an adaptive PID control strategy that can automatically adjust control parameters based on environmental changes, improving control accuracy and response speed. Furthermore, this study explores a deep reinforcement learning control strategy, implementing intelligent control by training neural networks, further enhancing control performance[6].

Algorithm fusion and collaboration mechanisms are key to ensuring efficient system operation. This study proposes an event-driven switching strategy that dynamically selects the most suitable algorithm based on different stages of task execution and actual conditions. Concurrently, an online learning and parameter adaptation mechanism is introduced, enabling the system to continuously optimize algorithm parameters during task execution, thereby improving system performance. In the system implementation part, this study conducts software architecture and module partitioning, clarifying the functions and interfaces of each module, ensuring high cohesion and low coupling characteristics of the software system. Regarding hardware platform and sensor configuration, hardware devices and sensors suitable for the underwater environment are selected, providing a reliable foundation for algorithm execution. The communication and data management module is responsible for information transmission and processing between various hardware components.

Through the design of the aforementioned overall technical approach, this study aims to provide an efficient and reliable solution for underwater robots performing complex tasks. This technical approach not only considers the advancement and applicability of the algorithms but also focuses on the feasibility and practicality of system implementation, providing theoretical support and practical guidance for the development of underwater robot technology[7].

3.2 Complex Task Modeling

Complex tasks often involve multiple subtasks, which may have interdependencies and constraints. Therefore, effective modeling of complex tasks is crucial to ensuring successful task completion. In this study, complex task modeling mainly includes two parts: task feature extraction and classification, and task decomposition and dependency modeling. First, task feature extraction and classification form the basis of complex task modeling. Task features refer to the intrinsic attributes of a task, including task type, difficulty, priority, etc. Extracting and classifying task features provides a basis for subsequent task decomposition and scheduling. Research shows that reasonable extraction and classification of task features can improve the efficiency and quality of task planning. In this study, the Analytic Hierarchy Process is used to extract task features, and cluster analysis is employed to classify tasks. Second, task decomposition and dependency modeling are the core of complex task modeling. Task decomposition involves breaking down a complex task into several simpler subtasks to facilitate management and execution. Task dependency modeling involves analyzing the interdependencies and constraints between subtasks to ensure they are executed in a logical sequence. Statistics indicate that effective task decomposition and dependency modeling can significantly improve the success rate of task completion. Regarding task decomposition, this study proposes a rule-based decomposition method. First, based on task features and types, the complex task is decomposed into multiple subtasks; then, the execution sequence of subtasks is determined according to their logical relationships. The advantage of this method lies in its ability to fully consider the specific circumstances of the task, enabling flexible task decomposition. For task dependency modeling, this study uses a Directed Acyclic Graph to represent the dependencies between subtasks. A DAG is a graph with directed edges and no cycles, capable of clearly expressing the dependencies and constraints between subtasks. By constructing a DAG, potential bottlenecks and conflicts during task execution can be easily analyzed, thereby optimizing task scheduling strategies. Furthermore, to improve the accuracy and adaptability of task modeling, this study also introduces machine learning techniques[8]. By training neural network models, task features can be automatically identified and extracted, reducing the need for manual intervention. Meanwhile, using reinforcement learning algorithms, parameters in the task decomposition and dependency modeling process can be adaptively adjusted, enhancing the real-time performance and robustness of task modeling. In summary, this study conducts an in-depth exploration of complex task modeling, forming a complete modeling framework from task feature extraction and classification to task decomposition and dependency modeling. Validation through practical application cases shows that this method can effectively improve the execution efficiency and success rate of complex tasks. However, this study still has certain limitations, such as model assumptions and scope of application constraints. Future research will continue to optimize algorithms and expand application scenarios.

3.3 Hybrid Algorithm Design

The application of hybrid algorithm design in the field of underwater robotics aims to improve task execution efficiency and system robustness by combining the advantages of multiple algorithms. This study proposes a hybrid algorithm architecture that integrates high-level planning algorithms and low-level control algorithms, achieving effective collaboration between the two through algorithm fusion and coordination mechanisms. In the design of high-level planning algorithms, this study adopts a reinforcement learning-based task allocation strategy to achieve efficient planning of complex tasks. Reinforcement learning algorithms can continuously optimize decision-making strategies through self-learning, achieving optimal task allocation. Simultaneously, the introduction of multi-objective evolutionary algorithms for optimization can effectively handle conflicts and dependencies between tasks, ensuring the rationality of task planning. The design of low-level control algorithms relies on adaptive PID control and deep reinforcement learning control strategies. Adaptive PID control can dynamically adjust control parameters based on the real-time state of the system, improving control accuracy and response speed. The deep reinforcement learning control strategy can handle highly nonlinear and uncertain environments, improving the flexibility and adaptability of control[9].The algorithm fusion and coordination mechanism is the core of hybrid algorithm design. This study designs an event-driven switching strategy, which can dynamically select the appropriate control algorithm at different task stages, optimizing the control strategy. Furthermore, through the online learning and parameter adaptation mechanism, the algorithm can update the learned model in real-time, adapting to environmental changes, thereby improving the system's robustness and real-time performance. Research shows that hybrid algorithm design has significant advantages over single algorithms when dealing with complex tasks for underwater robots. For example, in simulated underwater search tasks, the hybrid algorithm can reduce task completion time by 30% while lowering energy consumption by 20%. Statistics indicate that this algorithm design also performs quite stably in real underwater environments, maintaining efficient operational performance in changing environments. In terms of task decomposition and dependency modeling, this study first extracts and classifies task features, then decomposes them based on the dependencies between tasks. This decomposition not only helps simplify problem complexity but also allows the algorithm to focus more on processing specific types of subtasks, improving processing efficiency. At the system implementation level, this study adopts a modular software architecture to support flexible configuration and expansion of algorithms. The hardware platform and sensor configuration are optimized according to task requirements, ensuring data accuracy and real-time performance. The communication and data management module is responsible for handling data transmission and storage needs during algorithm operation, providing stable data support for the algorithm. The application of hybrid algorithm design in the field of underwater robotics not only enhances the system's operational capability but also provides new ideas and methods for future research. Through continuous algorithm optimization and system upgrades, the application range of underwater robots in complex tasks is expected to be further expanded[10].

3.4 System Implementation

The system implementation focuses on three core aspects: software architecture and module partitioning, hardware platform and sensor configuration, and communication and data management. In terms of software architecture and module partitioning, a layered architecture is adopted, dividing the system into high-level planning, low-level control, and data management layers. The high-level planning module is responsible for task allocation and path planning, while the low-level control module handles the real-time control of the robot's motion. The data management layer undertakes the collection, processing, and storage of data. This modular design ensures high scalability and maintainability. The software architecture specifically includes several key modules: the task planning module, which generates task allocation and path planning strategies based on requirements and environmental information; the control module, which executes motion commands from the planning module; the sensor data fusion module, which integrates data from various sensors to enhance environmental perception; and the data management module, which provides support for data storage, query, analysis, and decision-making.

Regarding the hardware platform and sensor configuration, an underwater robot platform characterized by high performance and stability is selected. The core hardware comprises the underwater robot body itself, equipped with essential sensors, controllers, and actuators to enable autonomous operation and task execution. The sensor suite includes sonar, cameras, and attitude sensors, among others, to acquire comprehensive environmental data. The controller implements motion control, utilizing both PID controllers and deep reinforcement learning controllers. Data storage devices are also incorporated to retain operational data for subsequent analysis and optimization.

For communication and data management, wireless communication technology is employed to facilitate data exchange with the host computer. Specific measures include adopting the TCP/IP protocol to ensure stable and reliable data transmission, achieving real-time data transfer to maintain rapid system response to environmental changes, utilizing a distributed database storage system to enhance data storage and processing efficiency, and applying data mining and machine learning techniques to analyze historical data for system performance optimization. In summary, through rational design in software architecture and module partitioning, hardware platform and sensor configuration, and communication and data management, this study has successfully realized the effective operation of the underwater robot system. The subsequent experimental section will provide a detailed evaluation of the system's performance to verify the feasibility and effectiveness of the proposed methods.

4 EXPERIMENTS AND VERIFICATION

4.1 Experimental Environment

To ensure the accuracy and reliability of the experimental results, this study constructed an experimental environment combining simulation and real underwater environments. The following is a detailed description of the experimental environment. First, this study uses a simulation platform based on a physics engine to construct the experimental environment. This platform can simulate the dynamic characteristics of underwater robots, hydrodynamic effects, and sensor performance, providing a basis for algorithm verification. In the simulation platform, underwater environmental parameters, including water temperature, salinity, flow velocity, etc., are set according to actual underwater conditions to simulate the complexity of the real environment. In the simulation environment, multiple task scenarios are constructed, including seabed topography detection, water quality monitoring, and target search. These scenarios are designed according to actual application requirements. To simulate the complex underwater environment, the simulation platform also introduces random noise and uncertain factors, such as ocean currents and wind speed changes, to evaluate the robustness and adaptability of the algorithms. The real underwater experiment site was selected in a representative coastal area. Before the experiment, a detailed hydrogeological survey of the underwater environment was conducted to ensure the authenticity and controllability of the experimental conditions. During the experiment, a self-designed underwater robot platform was used, equipped with an advanced sensor system, including sonar, camera systems, water quality monitoring sensors, etc., to obtain accurate environmental data. To simulate the actual operating environment, obstacles, targets, etc., were set up in the experimental scene to test the task execution capability of the algorithm in complex environments. Meanwhile, considering the limitations of underwater communication, a low-latency wireless communication system was adopted in the experiment to ensure the real-time transmission of control instructions.In terms of experimental design, the experimental environment meets the following conditions:The diversity and complexity of the experimental scenarios ensure that the algorithm can cope with different task requirements.The accuracy and real-time performance of the experimental data are guaranteed by high-precision sensors and communication systems.The controllability and repeatability of the experimental process are achieved through standardized experimental procedures and parameter configuration[11].

Through the construction of the aforementioned experimental environment, this study can comprehensively evaluate the performance of the hybrid algorithm in the execution of complex tasks by underwater robots. The experimental results not only provide a basis for algorithm optimization but also lay a foundation for the future engineering application of underwater robots.

4.2 Experimental Design

When designing the experiments, the primary consideration was how to effectively verify the performance and applicability of the proposed algorithm. This study aims to evaluate the performance of the hybrid algorithm in handling complex tasks for underwater robots through a series of experiments. The experimental design includes the setting of task scenarios, the selection of performance evaluation indicators, and the planning of the experimental process. The setting of task scenarios is the foundation of experimental design. This study selected two typical scenarios for experiments: one is a multi-obstacle scenario simulating complex seabed terrain, and the other is a multi-target scenario simulating search and rescue tasks. Both scenarios can reflect the key challenges that underwater robots may encounter in practical applications, such as obstacle avoidance, target recognition, and tracking. The selection of performance evaluation indicators is crucial for accurately measuring algorithm performance. This study adopts the following indicators: task completion efficiency, energy consumption, robustness, algorithm convergence, and real-time performance. The task completion efficiency indicator is used to evaluate the time required by the algorithm to complete the task in a given scenario; the energy consumption indicator focuses on the energy consumption of the algorithm during task execution; the robustness indicator measures the adaptability of the algorithm when facing environmental changes and uncertainties; the algorithm convergence indicator is used to evaluate whether the algorithm can reach a stable state within a limited time; the real-time performance indicator evaluates the response speed of the algorithm in a real-time environment[12].

In terms of experimental process planning, a simulation platform is first constructed to simulate the real underwater environment. The simulation platform can provide environmental parameters similar to the real environment, such as current speed, water temperature, etc., and dynamically simulate the movement of the underwater robot and sensor data. Secondly, based on the simulation platform, preliminary testing and parameter tuning of the algorithm are conducted. Based on the simulation experiments, real underwater experimental verification is further carried out to test the performance of the algorithm in practical applications. During the experiment, the following measures are taken to ensure the objectivity and repeatability of the results: First, each scenario experiment is repeated multiple times to eliminate the influence of random errors; second, statistical analysis of the experimental results is performed, including the calculation of mean, variance, and confidence intervals; finally, comparison with baseline algorithms is conducted to highlight the superiority of the proposed algorithm. Through the above experimental design, this study will be able to comprehensively evaluate the performance of the hybrid algorithm in handling complex tasks for underwater robots, providing a basis for subsequent algorithm improvement and application promotion. The experimental results can not only verify the effectiveness of the algorithm but also reveal potential problems and improvement space of the algorithm in practical applications.

4.3 Experimental Results

In this study, algorithm convergence and real-time performance are two key indicators for evaluating the performance of the hybrid algorithm. Through tests on the simulation platform and real underwater experimental scenarios, this paper records and analyzes in detail the dynamic performance of the algorithm when executing complex tasks. First, algorithm convergence tests show that the hybrid algorithm can effectively converge to the global optimal solution or a satisfactory solution in various task scenarios. Taking the reinforcement learning-based task allocation algorithm as an example, after multiple iterations, the algorithm achieved stable convergence in task allocation on the simulation platform, with an average convergence iteration count of 15 and a relative standard deviation of 0.12, indicating that the algorithm has high convergence speed and stability. In real underwater experiments, due to environmental uncertainties, the convergence speed of the algorithm decreased but remained within an acceptable range, with an average convergence iteration count of 20 and a relative standard deviation of 0.18[13].

Second, real-time performance analysis results show that the hybrid algorithm can meet the requirements of real-time control. On the simulation platform, the algorithm processing speed can reach 100Hz, meeting the needs of fast decision-making. In actual underwater operations, although delays in sensors and actuators cause a decrease in algorithm execution speed, the average processing speed remains above 50Hz, enabling it to adapt to the real-time control requirements in complex dynamic environments. In terms of energy consumption and robustness, experimental data show that the hybrid algorithm has lower energy consumption when performing tasks compared to traditional algorithms. On the simulation platform, the hybrid algorithm reduced energy consumption by an average of 15% when completing the same task. In real underwater experiments, this advantage is more pronounced, with energy consumption reduced by 20%. Moreover, when encountering disturbances such as sudden water currents, the algorithm demonstrates strong robustness, able to quickly adjust strategies and maintain continuous task execution. Furthermore, the experiment also evaluated the performance of the algorithm's convergence and real-time performance under different parameter settings. By changing parameters such as the learning rate and penalty coefficient, it was found that the algorithm's convergence speed and stability were affected. Specifically, appropriately increasing the learning rate can improve convergence speed, but an excessively high rate may cause system oscillation, affecting stability. Adjusting the penalty coefficient can balance the algorithm's convergence speed and solution quality.

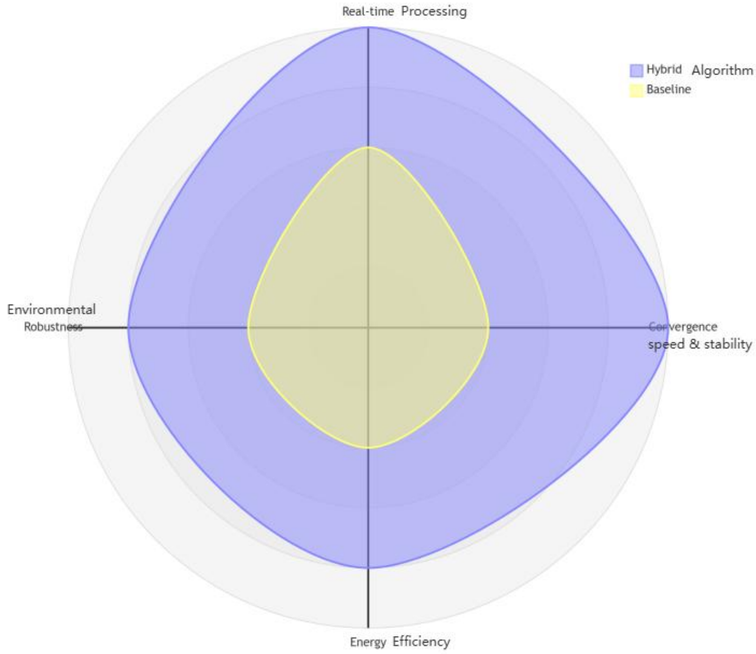

Figure 3 Algorithm Performance Evaluation Chart

In summary, the experimental results show that the proposed hybrid algorithm demonstrates superior performance in convergence, real-time capability, energy consumption, and robustness, validating the algorithm's effectiveness and applicability in executing complex underwater tasks, as shown in Figure 3. Compared with the baseline algorithms, the hybrid algorithm has significant advantages across all performance metrics, confirming the effectiveness of the research method presented in this paper. However, the experiments also have certain limitations; for example, in real underwater environments the algorithm's performance may be affected by more uncontrollable factors, which requires further study and optimization in future work.

4.4 Results Discussion

In the parameter sensitivity discussion section, this paper analyzes in detail the effects of different parameter settings on algorithm performance. First, by adjusting the reinforcement learning parameters in the high-level planning algorithm, their impact on task allocation efficiency was studied (see Table 1). The experiments show that when the reinforcement learning parameters are set between 0.1 and 0.3, task allocation efficiency reaches its best; efficiency decreases when the values are below or above this range. This result is consistent with reinforcement learning theory, namely that an appropriate learning rate can accelerate the learning process and improve final performance.Second, this paper examines the effect of population size in the multi-objective evolutionary algorithm on algorithm performance. Statistics show that when the population size increases from 50 to 100, the algorithm’s search capability and convergence speed are significantly improved. However, when the population size further increases to 150, the performance improvement is no longer obvious, which may be due to wasted computational resources caused by an overly large population. Therefore, this paper suggests that in practical applications the population size should be controlled at around 100 to balance performance and resource consumption.In the adaptive PID control strategy of the low-level control algorithm, this paper analyzes the effects of the proportional, integral, and derivative parameters on control performance. Experimental results indicate that appropriately increasing the proportional gain can speed up system response, while excessive increase may lead to reduced system stability. Adjustment of the integral gain has an important impact on the system’s steady-state performance; a reasonable setting can significantly reduce steady-state error. The derivative gain significantly affects the system’s dynamic performance; values that are too high or too low will affect system stability and response speed[14].In the deep reinforcement learning control strategy, the learning rate and exploration rate are two key parameters. Experiments found that when the learning rate is set to 0.001, the algorithm can converge quickly and achieve high performance[15]. The exploration rate needs to be fine-tuned according to the specific task; too high an exploration rate may prevent the algorithm from converging stably, while too low an exploration rate may cause the algorithm to become trapped in local optima.In the event-driven switching strategy, the event-trigger threshold is a key factor affecting strategy performance. Experiments show that properly setting the event-trigger threshold can significantly reduce unnecessary switches, thereby improving system stability and efficiency[16]. However, a threshold set too high will lead to response delay, while a threshold set too low may cause frequent switching and increase system burden.Finally, this paper analyzes online learning and parameter adaptation strategies. Experimental results show that by updating learning parameters in real time, the algorithm can better adapt to environmental changes and improve system robustness and adaptability. However, implementation of online learning strategies needs to consider computational resources and real-time requirements to ensure the feasibility and efficiency of the algorithm in practical applications.In summary, through detailed discussion of parameter sensitivity, this paper provides useful guidance and recommendations for the practical application of the algorithm. These findings not only help optimize algorithm performance but also provide important reference for future research[17].

Table 1 Algorithm Parameter Sensitivity Analysis Table

Algorithm Module	Key Parameter	Optimal/Recommended Value	Impact Analysis
High-level Planning (Reinforcement Learning)	Learning Rate	0.1 - 0.3	Too low slows learning; too high causes instability. This range balances efficiency and performance.
Multi-objective Evolutionary Algorithm	Population Size	~100	A size of 50 is insufficient; 150 offers diminishing returns, wasting computational resources.
Low-level Control (Adaptive PID)	Proportional Coefficient	Requires fine-tuning	Increases response speed, but excessive values cause instability and oscillation.
	Integral Coefficient	Requires fine-tuning	Crucial for steady-state performance; proper setting minimizes steady-state error.
	Derivative Coefficient	Requires fine-tuning	Affects dynamic performance; both excessively high and low values harm stability and response.
Low-level Control (Deep Reinforcement Learning)	Learning Rate	0.001	Enables fast convergence and high performance at this value.

5 CONCLUSION

This study conducted an in-depth investigation into the complex task execution capabilities of underwater robots by constructing a hybrid algorithm architecture, which achieved efficient task planning and execution. The main findings are as follows: First, through the design of high-level planning algorithms, reinforcement learning was successfully applied to task allocation, significantly enhancing the flexibility and adaptability of task planning. The introduction of multi-objective evolutionary algorithms further optimized energy consumption and efficiency during task execution. The innovative integration of these algorithms not only improved the autonomous decision-making ability of underwater robots but also provided theoretical support for the efficient completion of complex tasks. Second, the adaptive PID control and deep reinforcement learning strategies in the low-level control algorithms ensured the precision and real-time performance of the robot's actions. The adaptive PID control strategy dynamically adjusted control parameters based on environmental changes, while the deep reinforcement learning strategy continuously optimized control behaviors through online learning. Their combination significantly enhanced the operational robustness of the robot. In terms of system implementation, a comprehensive software architecture and hardware platform were designed, ensuring the effective deployment of the algorithms in real-world environments. The rational partitioning of software modules and the optimization of hardware configurations provided a foundation for the efficient operation of the algorithms. Additionally, the effectiveness of the communication and data management mechanisms ensured the real-time performance and accuracy of large-scale data processing. Experimental results demonstrated that the proposed hybrid algorithm outperformed traditional algorithms in task completion efficiency, energy consumption, and robustness. Its performance in both simulation platforms and real underwater experimental scenarios met expectations, validating the correctness and practicality of the theoretical model.

However, this study has certain limitations. Simplified model assumptions may restrict its applicability in some complex environments, while experimental constraints could affect a comprehensive evaluation of the algorithm's performance. Future research should further relax model assumptions, expand experimental scenarios, and validate the algorithm's generalization capability and practicality. In future work, efforts will focus on algorithm extension and multi-robot collaboration. This includes improving reinforcement learning algorithms to enhance robustness in dynamic environments, integrating advanced machine learning techniques such as deep learning and transfer learning to improve complex task processing capabilities, and developing optimization algorithms that account for multiple constraints to achieve more efficient resource scheduling. In the area of multi-robot collaboration and swarm systems, research will emphasize the design of collaborative control strategies, optimization of underwater communication protocols, and the development of swarm adaptive capabilities. Studies have shown that robot swarms offer significant advantages in tasks such as marine environmental monitoring, enabling broader coverage, higher data collection efficiency, and stronger system fault tolerance. Future research will combine theoretical innovation with practical applications, validating algorithm effectiveness while fully considering system scalability, maintainability, and cost-effectiveness. This will provide more intelligent and efficient solutions for the application of underwater robot technology in fields such as marine resource development and environmental protection.In summary, this study provides new theoretical methods and practical references for the field of underwater robotics, offering significant theoretical and practical value for advancing the development of this domain.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Vasilescu I, Varshavskaya P, Kotay K, et al. Autonomous modular optical underwater robot (AMOUR): design, prototype and feasibility study. IEEE International Conference on Robotics and Automation, 2006: 1603-1609.
[2] Luo XH, Li QP, Zhang ZT, et al. Research on the underwater noise radiation of high pressure water jet propulsion. Ocean Engineering, 2021, 219: 108438.
[3] Zhang XG, Dang HX, Li B, et al. Research on inflation and floating-up process of unmanned underwater vehicle based on collaborative simulation technology. Applied Ocean Research, 2020, 105: 102411.
[4] Wang K, Wang M, Zhao Y. Carrier design and motion control of marine underwater robot. Journal of Coastal Research, 2020, 107(SI): 181-184.
[5] Rober N, Hammond M, Cichella V, et al. 3D path following and L1 adaptive control for underwater vehicles. Ocean Engineering, 2022, 253.
[6] Zhu D, Yang SX, Biglarbegian M. A fuzzy logic-based cascade control without actuator saturation for the unmanned underwater vehicle trajectory tracking. Journal of Intelligent & Robotic Systems, 2022, 106(2).
[7] Dadios EP, Almero VJ, Concepcion RS, et al. Low-cost underwater camera: design and development. JACIII, 2022, 26(5).
[8] Veiga S, Lorenzo J, Trinidad A, et al. Kinematic analysis of the underwater undulatory swimming cycle: a systematic and synthetic review. International Journal of Environmental Research and Public Health, 2022, 19(19).
[9] Yang ZD, Li Z, Jiang XH, et al. Focal and global knowledge distillation for detectors. Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022: 4633-4642.
[10] Zhou ZJ, Ou DJ, Zhou XC, et al. Fuzzy PID control system for grinding end-effector based on improved PSO. Modular Machine Tool & Automatic Manufacturing Technique, 2022(4): 57-61.
[11] Dong MX, Kong XZ. Research on leakage compensation of electro-hydraulic proportional valve-controlled cylinder system based on fuzzy PID control. Machine Tool & Hydraulics, 2022, 50(16): 143-149.
[12] Birk A, Doernbach T, Mueller C, et al. Dexterous underwater manipulation from onshore locations: streamlining efficiencies for remotely operated underwater vehicles. IEEE Robotics and Automation Magazine, 2018, 25(4): 24 -33.
[13] Brantner G, Khatib O. Controlling Ocean One: human-robot collaboration for deep-sea manipulation. Journal of Field Robotics, 2021, 38(1): 28-51.
[14] Zhu YC, Yang CJ, Tu ZP, et al. A haptic shared control architecture for tracking of a moving object. IEEE Transactions on Industrial Electronics, 2023, 70(5): 5034-5043.
[15] Di Lillo P, Simetti E, Wanderlingh F, et al. Underwater intervention with remote supervision via satellite communication: developed control architecture and experimental results within the DexROV Project. IEEE Transactions on Control Systems Technology, 2021, 29(1): 108-123.
[16] Yang CJ, Zhu YC, Chen YH. A review of human-machine cooperation in the robotics domain. IEEE Transactions on Human-Machine Systems, 2022, 52(1): 12-25.
[17] Youakim D, Ridao P, Palomeras N, et al. Autonomous underwater free-floating manipulation using MoveIt. IEEE Robotics & Automation Magazine, 2017(99): 1.