Journal of Computers Vol. 37 No. 5, October 2026, pp. 339-355
https://doi.org/10.63367/19911599202610370502 1

DDS-Based Perception and Control System for Plateau Obstacle Removal Vehicles

Meihui Chen[1], Zhenjiang Zhang[1*], Ling Zhou[2], and Yawei Lv[1]

[1] School of Electronic Information Engineering, Beijing Jiaotong University, Beijing, 100044, China

{25120142, zhangzhenjiang, 25120194}@bjtu.edu.cn

[2] Information Technology Institute, China Railway Nanchang Group Co., Ltd., Nanchang 330003, Jiangxi, China

1499547675@qq.com

Received 10 July 2026; Revised 29 July 2026; Accepted 10 September 2026

Abstract. To overcome the challenges of ensuring real-time communication, insufficient fusion of heterogeneous sensing data, and limited edge computing resources for high-altitude obstacle removal vehicles operating in extreme environments, this paper proposes a distributed software architecture using a Data Distribution Service (DDS) middleware. First, a QoS-aware software layering model is created, employing different QoS policy configurations to achieve deterministic scheduling and isolation of mixed critical data streams. Second, a mechanism using an improved particle swarm optimization algorithm to schedule heterogeneous computing resources addresses the load balancing problem of multimodal sensing and control tasks across heterogeneous cores. Third, a multi-source heterogeneous sensing data encapsulation and fusion service based on a standardized Interface Definition Language (IDL) is established, and an intrinsic security protection system is built using the DDS Security specification. Experimental results show that this architecture can effectively reduce the transmission latency and jitter of critical control commands, improve the throughput and robustness of the system under high concurrency environments, and provide a feasible technical approach for the development of intelligent software for special vehicles.

Keywords: high-altitude obstacle removal vehicle, distributed software architecture, data distribution service, quality of service, edge computing, resource scheduling

1 Introduction

1.1 Research Background

My country's vast plateau regions, with their unique geological structures and harsh climates, have long been plagued by severe natural disasters. The complex geological structure of these regions, coupled with the interplay of permafrost changes, groundwater movement, and human engineering activities, results in poor road surface stability and a high risk of collapse. The high altitude, low oxygen, low temperature, and strong ultraviolet radiation further exacerbate the suddenness and destructive nature of these disasters, posing significant challenges to emergency response and road repair. Road collapses are a common geological hazard, occurring frequently and severely impacting local traffic safety, economic development, and residents' lives. Traditional disaster relief methods largely rely on manual operation and basic mechanical equipment, which are not only inefficient but also pose significant risks to personnel safety and equipment reliability in the high-altitude, oxygen-deficient environment, failing to meet the need for rapid traffic restoration.

With the development of new infrastructure and smart transportation strategies, replacing manual labor with intelligent unmanned equipment in high-risk areas has become an inevitable trend. The new generation of plateau obstacle removal vehicles is no longer a simple aggregation of mechanical power; it is a mobile robot integrating high-performance computing, multi-source sensing, and intelligent decision-making. However, there are still many shortcomings in the transformation of plateau obstacle removal technology towards intelligence.

* Corresponding Author

Traditional equipment mostly adopts mechanical hydraulic system design, which often results in insufficient power, slow response and reduced operation accuracy in the extreme environment of plateau. The false alarm rate of sensors will increase in strong light or haze. Obstacle recognition mainly relies on a single sensor, which is easily affected by the environment in complex terrain, resulting in missed detection or misjudgment, which greatly increases the risk of operation.

In order to solve the problems mentioned above, the vehicle electronic and electrical architecture (E/E architecture) is undergoing profound changes from distributed to domain-centralized or even central computing. In this process, software-defined vehicles (SDV) have emerged. For plateau obstacle removal vehicles, this means that the core capabilities of the vehicle are determined by software, rather than just relying on hardware stacking. However, with the addition of high-bandwidth sensing devices such as lidar, high-resolution cameras, interferometric synthetic aperture radar (InSAR), and distributed fiber optic sensors, the on-board data traffic has increased exponentially. This makes plateau obstacle removal vehicles mobile edge computing nodes working in extreme environments [1].

In this case, traditional on-board network protocols and closed software structures can no longer meet the requirements for real-time transmission and processing of massive heterogeneous data. In situations of scarce computing resources, unstable network communication, and severe environmental interference, how to create a highly reliable, low-latency, and adaptive software system has become a current research hotspot. As Merino et al. mentioned in their research on distributed secure edge computing platforms, it is very important to use modular Internet of Things (IoT) hardware platforms and distributed computing architectures to solve data processing bottlenecks in extreme environments [1].

The special feature of plateau obstacle removal operations is that they have two extreme requirements for real-time performance and accuracy. On the one hand, vehicles need to detect, classify, and plan trajectories of obstacles within milliseconds; on the other hand, the control system needs to ensure the dynamic stability of vehicles on nonlinear terrain and low-adhesion surfaces. Existing single-sensor mode cannot cope with the complex and changeable lighting and meteorological environment of plateaus. For example, visual sensors will fail when there is strong reflection in snow, and lidar will have increased noise in sand and dust. Therefore, environmental perception technology based on multi-source information fusion is very important.

Moreover, the complexity of control algorithms is also constantly increasing. In order to overcome the shortcomings of traditional PID (proportional-integral-derivative) algorithm in plateau nonlinear working conditions such as poor adaptability and response lag, advanced algorithms such as model predictive control (MPC) and reinforcement learning (RL) have been used in vehicle control systems. However, the high computational load of these algorithms poses a huge challenge to the resource scheduling capability of the vehicle computing platform. Without effective middleware-level resource management and QoS guarantee, high-priority control commands can easily be blocked by low-priority sensor data streams, which can lead to serious safety accidents [2]. Therefore, taking data-centric data distribution service as the communication backbone and creating a software architecture that integrates perception, decision-making and control is an inevitable choice to improve the intelligence level of plateau obstacle removal vehicles.

1.2 Current Situation Analysis

Regarding the intelligent system of plateau obstacle removal vehicles, the academic and industrial communities are currently focusing on four aspects: environmental perception algorithms, complex system control, communication middleware architecture, and network security protection.

In the early stages of environmental perception, traditional digital image processing techniques were mainly used. Konrad and Bozinovic conducted a detailed analysis of uniform translation of image motion and compared the advantages and disadvantages of DCT and FT transformation methods in image analysis, laying a theoretical foundation for early visual speed measurement and obstacle detection [3]. Gil-Jimenez et al. proposed a method to classify traffic sign shapes using support vector machines (SVM) and speckle feature FFT transformation, which is a typical application of classical machine learning in structured road perception [4].

However, when encountering complex obstacles such as rocks, potholes, and snow on unstructured roads in plateau regions, traditional methods cannot guarantee their robustness. In recent years, the explosion of deep learning technology has brought breakthroughs to this field. Li et al.'s attention mechanism RNN was used to identify radar pulse repetition interval (PRI), enabling the radar system to have stronger anti-interference capabilities against complex electromagnetic environments [5]. Similarly, Ramasamy et al.'s research approach of using RNNs to classify plant nutrient deficiencies has also been applied to the identification of geological

features based on hyperspectral images, demonstrating the advantages of temporal neural networks in processing continuous environmental features [6]. In addition, Ye et al.'s feasibility study on the spectral detection of bovine serum albumin, although a study in the field of bio-optoelectronics, can provide interdisciplinary reference for the remote sensing detection of special minerals or geological components on the plateau [7]. In order to make perception more comprehensive, multimodal data fusion has become the mainstream trend. Wang et al. proposed a dynamic clustering recommendation algorithm based on a two-layer graph attention network. This method uses the idea of graph neural networks (GNN) to aggregate data and has been widely applied to the heterogeneous data fusion of laser point clouds and visual images, which effectively solves the problem of single sensor failure under severe weather conditions [8].

Vehicle control and trajectory planning in plateau environments are characterized by low adhesion, large slope, etc., and vehicle dynamics have strong nonlinearity. Traditional linear control methods cannot guarantee control accuracy. Bekiaris-Liberis and Vazquez proposed a nonlinear bilateral full-state feedback trajectory tracking control method for a class of viscous Hamilton-Jacobi partial differential equations (PDEs), theoretically solving the boundary control problem of distributed parameter systems and providing a mathematical tool for precise trajectory tracking of flexible robotic arms or long-wheelbase vehicles [9]. To cope with external environmental disturbances, Patelski and Pazderski proposed a parameter identification disturbance suppression control method with asymptotic error convergence characteristics, which plays a crucial role in maintaining vehicle operational stability under strong winds at high altitudes [10].

With the development of artificial intelligence technology, data-driven control methods are becoming increasingly advantageous. Zheng et al. established an electronic fuel injection system for agricultural hybrid unmanned aerial vehicles using artificial intelligence technology, optimizing the air-fuel ratio with intelligent algorithms to effectively improve engine energy efficiency under various operating conditions. This provides a direct reference for solving the problem of reduced diesel engine power in hypoxic environments at high altitudes [11]. In addition, regarding multi-agent cooperative operations, Zhao et al. proposed the MCMARL framework, which uses a hybrid classification distribution parameterized value function to solve the problem of action space dimension explosion in multi-agent reinforcement learning, providing algorithmic support for the cooperative formation and task allocation of future plateau obstacle removal vehicle groups [12].

In terms of energy management, Jiang et al., taking into account the risk of seasonal power imbalance, proposed a method for renewable power system planning, which is instructive for the design of energy management strategies for plateau obstacle removal vehicles using hybrid or pure electric drive during long-term field operations [13].

With the increasing number of onboard sensors, how to efficiently transmit and schedule large amounts of data has become a bottleneck in system design. Traditional point-to-point communication or bus broadcast modes are insufficient in terms of bandwidth utilization and real-time guarantee. Zhang et al. proposed a general k-anycastDTN routing algorithm to solve the data delivery problem in intermittently connected networks, which has high application value for plateau areas with insufficient communication infrastructure [14].

In order to achieve efficient service orchestration under limited resources, Magedanz and Schreiner studied QoS-aware multi-cloud agents for next-generation network (NGN) services and concluded that elastic resource allocation can achieve good performance benefits [2]. Applying this idea to the vehicle software architecture, the QoS policy configuration of data distribution services based on DDS was developed. DDS adopts a data-centric publish-subscribe model, decoupling space and time, and can provide system designers with different QoS parameters to configure reliability, latency budget, etc.

From a software engineering perspective, in order to cope with the increasingly complex variant management of vehicle software, the research of Michelon et al. shows that the problems caused by feature update propagation are very obvious in preprocessor-based software product lines, and believes that modular and component-based design patterns are of great significance for reducing software maintenance costs and improving system development capabilities [15]. The I-Drive modular system architecture of Zulkepli et al. provides a standard for hardware and software interfaces of intelligent vehicle research platforms, providing an engineering paradigm for rapid prototyping [16].

With the increasing degree of vehicle networking, network security threats cannot be ignored. Comaneci, Dobre, and others used software-defined networking (SDN) and machine learning to ensure network security, believing that separating the control plane and data plane and using traffic anomaly detection algorithms can effectively defend against DDoS attacks and man-in-the-middle attacks [17]. Kirubakaran et al. conducted a detailed analysis of honeypot technology at different levels, designed a real-time honeypot system, and adopted the idea of proactive defense to lure malicious scans of vehicle networks, thereby protecting the security of the core control unit [18].

Yuan and Ye conducted a detailed analysis of the energy consumption and faults of data center power systems, proposed a fault correlation model and energy efficiency evaluation index, which can be transferred to the health management (PHM) system of the on-board computing platform of the plateau obstacle removal vehicle to prevent software crashes caused by power supply problems [19]. Causal inference technology has also been used in the fault diagnosis of software systems. Liu et al. used the G-Computation method to estimate the causal effects of sometimes variable confounding factors, providing a new statistical means for finding cascading faults in complex software systems [20].

Hardware modularization and platformization have laid the foundation for the implementation of software architecture. Discussions on hardware platformization and hardware modularization in the domestic CSDN technical community suggest that standard interface design is a prerequisite for software and hardware decoupling [21]. The "passenger-type plateau turbocharged vehicle" reported by Sina Technology is designed for the special environment of plateau, and the hardware has already achieved corresponding adaptation results, providing a mature vehicle platform for unmanned transformation [22].

Finally, considering the requirements of human-machine collaborative operation, operator training is very important. Sitthiworachart studied the application of the flipped classroom method in cultivating students' thinking ability and improving students' participation. Although it is mainly from an educational perspective, the concept of active learning and interactive training reflected therein has reference value for designing an efficient remote driver training system and reducing the cognitive burden of human-machine interaction [23]. The existing centralized control architecture has some shortcomings.

Although intelligent technology has been applied to special vehicles, most of the current equipment still uses a centralized electronic and electrical architecture based on the controller area network (CAN) bus or a single central computing unit (CCU). In a special scenario like plateau obstacle removal, traditional siloed or hard-coupled software architectures have gradually exposed many insurmountable technical bottlenecks.

The bottleneck effect of computing and communication bandwidth is becoming increasingly prominent. In a centralized architecture, all raw data from sensors must be aggregated to the central controller for processing. Due to the increased density of lidar point clouds and the resolution of visual sensors, the concurrent transmission of a large amount of unstructured data can easily exhaust bus bandwidth, resulting in "data congestion". When high-priority obstacle avoidance braking commands and low-priority environmental log data compete for the transmission channel, the end-to-end latency of control commands is unpredictable due to the lack of fine-grained traffic shaping. Studies have shown that in nonlinear systems with random network interference, the randomness of communication latency makes it difficult for trajectory tracking errors to converge, which is one of the main reasons [2].

The risk of single point of failure (SPoF) is high, and the system's fault tolerance is insufficient. Centralized software systems are typically organized with tightly coupled code structures, meaning that perception, planning, and control modules run in the same runtime environment. If the central processing node's operating system experiences a kernel crash or memory leak, the entire vehicle loses control, especially in harsh environments like high-altitude regions where rescue is costly. Distributed architectures, by breaking down functionality into different computing nodes, can effectively isolate fault domains and prevent localized software failures from causing a cascading effect across the entire system.

There is a lack of isolation and scheduling capabilities for mixed-criteria tasks. High-altitude obstacle removal operations involve various task flows with different characteristics, some requiring high real-time performance (Hard Real-Time) and others requiring high throughput (Best-Effort). Existing general-purpose operating systems or simple real-time operating systems (RTOS) scheduling strategies struggle to simultaneously meet these two completely different QoS requirements on a single computing platform. The lack of a priority-based dynamic resource scheduling mechanism makes it easy for critical tasks to be "starved" or "deadline violated" when the system encounters sudden computational loads, thus affecting the safety and continuity of obstacle removal operations.

1.3 Explanation of the Limitations and Problems of Existing Technologies

While the aforementioned research has achieved significant progress in its respective fields, integrating them into the specific environment of a high-altitude obstacle removal vehicle still presents numerous problems and shortcomings.

Data silos and the conflict between real-time performance are quite evident. The existing architecture develops perception, decision-making, and control modules as independent systems, lacking a unified data

exchange standard. When high-bandwidth visual data and high-frequency control commands are transmitted simultaneously over the network, the absence of fine-grained QoS management easily leads to latency and jitter in critical control data, thus impacting the safety of the vehicle during high-speed operations. To address this issue, a communication middleware architecture, as shown in Fig. 1, that provides fine-grained QoS guarantees is needed.

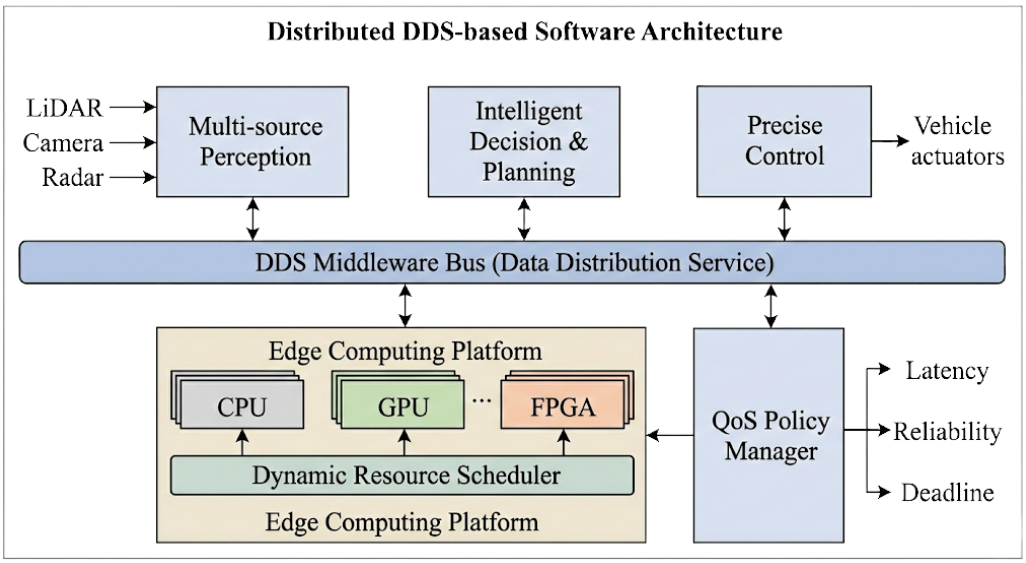


Fig. 1. Distribute DDS software architecture

There is a lack of dynamically scheduled edge computing resources. Plateau obstacle removal vehicles are mobile edge nodes with very limited computing resources (CPU/GPU/memory) that are easily affected by ambient temperature. Existing static resource allocation methods cannot handle sudden increases in computational load during obstacle removal tasks, easily causing system lag or crashes.

The degree of fusion of heterogeneous multi-source sensing is low. Although multi-sensor fusion has become a consensus, current fusion is mostly performed in the post-processing stage, lacking effective fusion algorithms at the raw data layer or feature layer, and failing to fully exploit the data potential of new sensors such as fiber optic sensing and InSAR.

Network security protection systems have vulnerabilities. Cyberattacks against special vehicles are becoming increasingly sophisticated, but existing in-vehicle networks lack inherent security mechanisms such as encrypted transmission, authentication, and abnormal traffic scrubbing, leaving them vulnerable to malicious hijacking.

1.4 Research Content and Innovations of This Paper

To address the above issues, this paper primarily studies the design and implementation of a software-based architecture using DDS middleware. The main research content and innovations of this paper are fourfold, as detailed in the table below.

Propose a QoS-aware distributed DDS software architecture. This paper breaks away from the traditional centralized control method, adopting a publisher-subscriber-based model to establish a distributed structure. By defining the Quality of Service (QoS) policy in the fault-clearing service scenario, deterministic scheduling of data streams with different priorities is achieved, ensuring that critical control commands arrive in a timely manner during network congestion, thus improving system robustness.

Design a dynamic scheduling algorithm for heterogeneous computing resources based on edge computing. This paper encapsulates sensing and control tasks using containerization technology and proposes a dynamic scheduling mechanism using an improved Particle Swarm Optimization (PSO) algorithm. This mechanism adjusts the allocation of tasks across heterogeneous cores (CPU, GPU, FPGA) according to the vehicle's current

operating conditions and the load of computing units, thereby maximizing the throughput of the entire system and overcoming the computing bottleneck caused by resource scarcity.

Create a method for spatiotemporal synchronization and fusion of multi-source heterogeneous sensing data A unified data abstraction layer and time synchronization protocol were designed for LiDAR, vision cameras, and distributed fiber optic sensors. Based on this, a graph neural network (GNN) was used to fuse and deepen the feature layers of heterogeneous data, improving the accuracy of obstacle recognition and the reliability of environmental perception in harsh environments such as high altitudes, strong sunlight, and snowstorms. (See Fig. 2.)

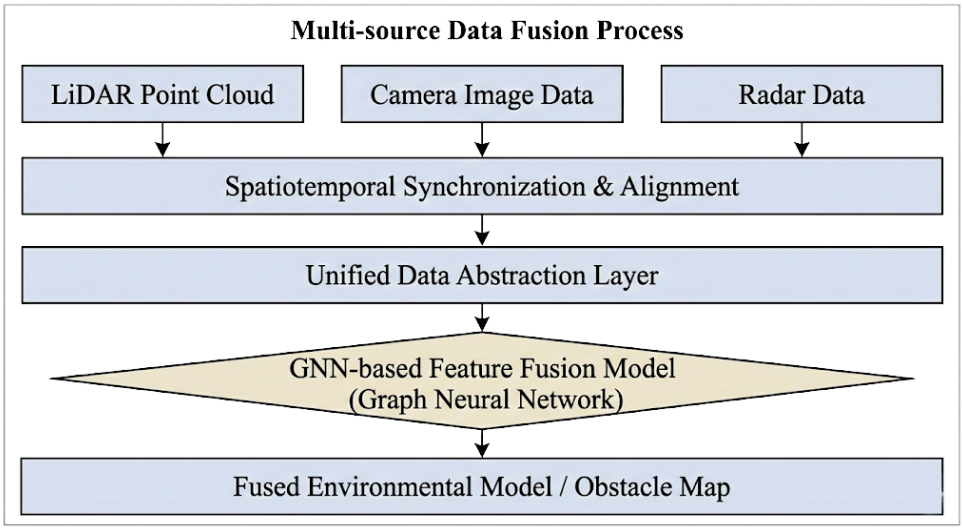


Fig. 2. Multi-source data fusion process

This project aims to create a proactive defense-oriented network security system based on the principles of Software-Defined Networking (SDN). Adopting the SDN architecture, it incorporates traffic monitoring and encryption modules into the communication middleware layer. Lightweight machine learning algorithms are used for real-time detection of abnormal traffic patterns, honeypot technology is employed to lure potential intruders, and a defense-in-depth system is established to ensure the information security of unmanned equipment.

2 Distributed Software Architecture Design for Plateau Barrier Removal Vehicle Based on DDS

2.1 System Overall Logical Architecture and Domain Controller Partitioning

To meet the high reliability, real-time performance, and scalability requirements of high-altitude obstacle removal vehicles operating in extreme environments, this paper abandons the traditional centralized electronic and electrical architecture and proposes a distributed, hierarchical software architecture based on a data distribution service middleware. This architecture design follows the principles of high cohesion and low coupling, using Service-Oriented Architecture (SOA) to break down complex obstacle removal tasks into independent autonomous functional units, achieving efficient collaboration through standardized communication interfaces. This not only solves the computing power and bandwidth bottlenecks encountered in traditional architectures but also significantly improves system fault tolerance and the flexibility of upgrades and maintenance.

The overall logical architecture of the system can be divided into four layers from bottom to top: the basic hardware layer, the system software layer, the communication middleware layer, and the application function layer. The basic hardware layer consists of a heterogeneous computing platform, a multi-source sensor array,

and actuators. The system software layer includes a real-time operating system (RTOS) and a board support package (BSP), providing a stable operating environment and hardware abstraction for upper-layer applications. The communication middleware layer is the hub of the entire architecture, using DDS as a unified data exchange backbone network. DDS employs a data-centric publish-subscribe model, shielding the differences in underlying network protocols and hardware platforms, and providing a virtual, globally shared data space for application modules distributed across various computing nodes. The application functional layer is divided into several functional domains according to business logic. Software modules within each functional domain interact with each other using the DDS interface to perform functions such as environment awareness, path planning, and job control. The overall logical architecture of the system is shown in Fig. 3 below.

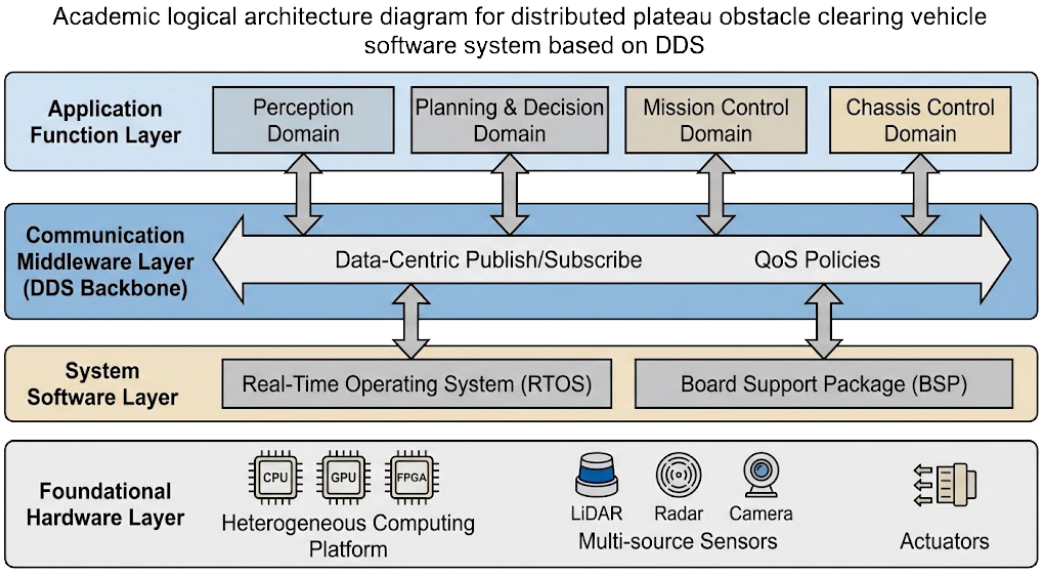


Fig. 3. Academic logical architecture diagram of the distributed plateau obstacle clearing vehicle software system based on DDS

2.2 DDS Communication Middleware Model Design

In the distributed software architecture proposed in this paper, the data distribution service serves as both a communication protocol and a core data model and interaction principle. The DDS middleware model is designed to create a data-centric, globally shared information space, completely breaking down the separation between data producers and consumers. This effectively addresses the requirements of high-altitude obstacle removal vehicles for real-time sharing and collaborative control of multi-source information in complex operating environments.

The key to the DDS model is its data-centric publish/subscribe (DCPS) mechanism. In this layer, each functional module in the system, such as the perception algorithm, path planner, or chassis controller, is treated as a domain participant. Participants join the same logically defined "global data space" through the DDS middleware. Data interaction no longer relies on point-to-point connections or fixed server addresses, but instead uses named "topics". A topic is the basic unit of data interaction; it defines the logical meaning and structure of the data. To ensure compatibility between the data types of different topics, the IDL interface definition language is used to define all topic data types, guaranteeing strict consistency across different byte orders and compiler environments. A topic called 'VehicleMotionState' is defined. The IDL structure of this topic contains strongly typed fields such as timestamp, 3D position coordinates, velocity vector, and heading angle. The perception module acts as the publisher, creating a corresponding topic data writer to send data to the global data space and announce its interest in the topic. The planning module acts as the subscriber, creating a data reader to indicate its interest in the topic. The DDS middleware automatically distributes matching data to all subscribers.

To address the coexistence of mixed critical tasks in high-altitude obstacle removal vehicles, a key aspect of the DDS model design is the fine-grained configuration of the Quality of Service (QoS) policy. The QoS policy is a powerful metadata set that can precisely control data transmission behavior to meet the different requirements for reliability, real-time performance, and persistence in various business scenarios. For high-frequency, high-volume sensor data streams, such as LiDAR point clouds or camera video frames, real-time performance is much more important than data integrity; losing some older data frames is perfectly acceptable. Therefore, for this type of topic, the reliability QoS policy is set to best-effort (BEST_EFFORT) to reduce transmission latency and save network bandwidth. Control commands related to vehicle safety, such as emergency braking and steering angle settings, can cause serious traffic accidents if data is lost. For this type of issue, a reliability strategy needs to be configured to be reliable, and a history QoS (Quality of Service) setting needs to be implemented to ensure that the middleware can automatically retransmit data until it is acknowledged to have been received during network fluctuations. In addition, to ensure the determinism of the control loop, a deadline QoS strategy is added, requiring data to be updated within a specified time window. If the timeout occurs, a notification mechanism is triggered, allowing the system to take appropriate fault response measures. Using policy-driven design, the DDS middleware successfully creates multiple logically isolated data channels with different service levels on a single physical network. The conceptual model of the DDS communication middleware is shown in Fig. 4.

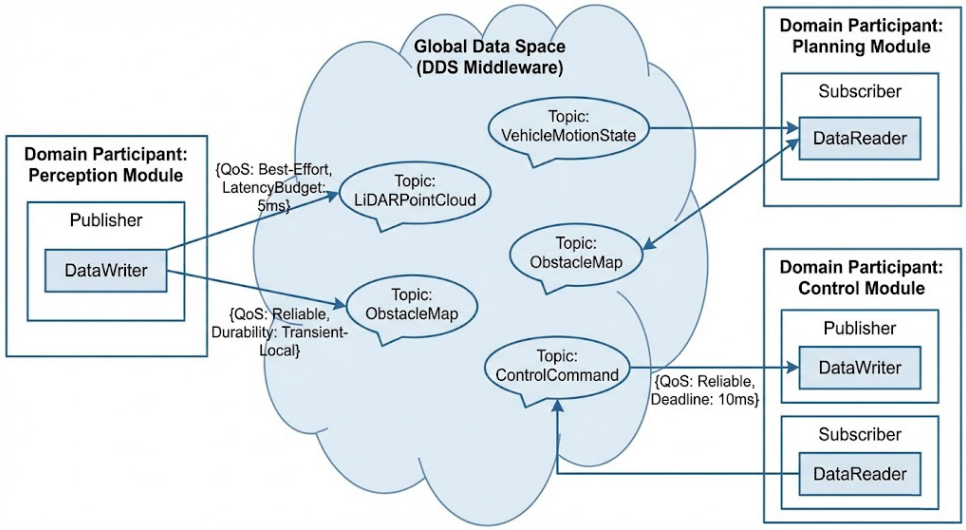


Fig. 4. Conceptual model diagram of DS communication middleware

2.3 QoS Policy Configuration for Troubleshooting Tasks

The advantage of DDS middleware lies in its diverse Quality of Service (QoS) control strategies, allowing for personalized data transmission settings based on data business attributes and system resource availability. In the distributed software architecture of high-altitude obstacle removal vehicles, various data streams exist, each with different requirements for real-time performance, reliability, persistence, and bandwidth. Using only one transmission strategy would lead to resource waste or critical task failure. Therefore, different QoS strategies are adopted for obstacle removal tasks to ensure the system functions normally in extreme environments.

Based on the characteristics of high-altitude obstacle removal operations, the data streams in the system are roughly divided into three categories: high-bandwidth real-time sensing data, important control command data, and system status information data. Different QoS configuration sets are designed for each of these three data types.

For high-bandwidth real-time sensing data such as LiDAR point clouds and camera video streams, the data volume is large and the update frequency is high. The latest data frames are more sensitive, and the loss of some historical frames generally does not affect the overall environmental modeling. Therefore, for this type of data, the reliability strategy is set to best-effort to avoid additional network overhead and latency caused by

retransmission. Simultaneously, a low-latency budget is set so that the middleware can send such data first, ensuring the timeliness of the perceived information.

Conversely, for critical control command data such as chassis drive and robotic arm movements, although the data volume is small, the requirements for reliability and determinism are very high. Any lost or timed-out command will cause vehicle loss of control or operation failure. Therefore, the reliability strategy for this type of data must be set to "reliable" to ensure that the data is delivered to subscribers correctly. Furthermore, to ensure the real-time performance of the control loop, a strict deadline strategy is used to force data publishers to update data within a specified time. If a timeout occurs, the subscriber will receive a notification and trigger a safety degradation strategy.

For system status information data such as global obstacle maps and task planning paths, due to the low update frequency but the need for long-term storage, a high storage density is required. To ensure that newly joined nodes in the network can immediately obtain the latest system status, the persistence strategy is set to transient local. The DDS middleware caches recently published data samples locally, and automatically begins replaying historical data when a new subscriber joins. Through the aforementioned differentiated QoS configurations for specific tasks, the DDS middleware creates multiple logical data channels with different service levels on an unreliable physical network, ensuring deterministic transmission of critical control data and maximizing the utilization of limited network bandwidth. This lays a solid communication foundation for the stable and efficient operation of plateau obstacle removal vehicles.

3 Multi-source Sensing Data Stream Processing Services and Fusion Mechanisms

3.1 Standardized Packaging and Distribution of Heterogeneous Sensor Data

Based on the distributed architecture of DDS, the environmental perception capability of plateau obstacle removal vehicles can no longer be achieved by a single sensor, but rather relies on the collaborative work of multimodal sensors. To address the risk of sensor failure caused by the complex terrain and extreme climate of plateaus, vehicles are equipped with various heterogeneous sensors, such as LiDAR, high-resolution cameras, millimeter-wave radar, and inertial navigation systems. However, these sensors differ significantly in physical interfaces, data formats, sampling frequencies, and communication protocols, posing a major challenge to the unified processing of upper-layer fusion algorithms. Therefore, creating an efficient and standardized multi-source perception data stream processing service and fusion mechanism is essential for achieving accurate environmental modeling.

To shield the differences in underlying sensor hardware and decouple the perception layer from the decision layer, this paper establishes a sensor data standardization encapsulation layer on top of the DDS middleware. The main task of this layer is to convert the various proprietary format data streams from different hardware interfaces into a unified data structure conforming to OMG IDL (Interface Definition Language), add accurate timestamps and coordinate system information, and finally publish it to the entire data space in the form of DDS topics.

The standardization encapsulation process mainly includes the following key steps:

First, hardware abstraction and driver adaptation. Based on the different characteristics of various sensors, corresponding hardware drivers are written to read the raw binary data stream using physical interfaces. The LiDAR driver receives data packets via UDP sockets, while the millimeter-wave radar driver reads CAN frames via the SocketCAN interface.

Next is protocol parsing and data normalization. Raw data typically contains manufacturer-defined header information, checksums, and specially encoded measurement values. The parsing module strips away these protocol layers, retaining only the core measurement data, and then converts it all to internationally recognized standard units (SI units), such as converting distance to meters (m) and speed to meters per second (m/s). Furthermore, for subsequent multi-sensor spatial fusion, the installation positions and attitudes of all sensors must be referenced to the vehicle's base coordinate system (Base_link Frame), and the corresponding coordinate system identifier (Frame ID) must be included when publishing the data.

Finally, there is IDL structure encapsulation and DDS publishing. Based on a predefined standard IDL file, the normalized data is filled into the corresponding data structure. To ensure time synchronization of multi-sensor data, each data sample must include a high-precision acquisition timestamp generated from a unified clock source. After encapsulation, the data writer publishes the standardized data samples to the designated DDS

topic according to the QoS policy defined in Section 2.3, making them available for subscription by downstream fusion and planning modules. The flowchart for the standardized encapsulation and publication of heterogeneous sensor data is shown in Fig. 5.

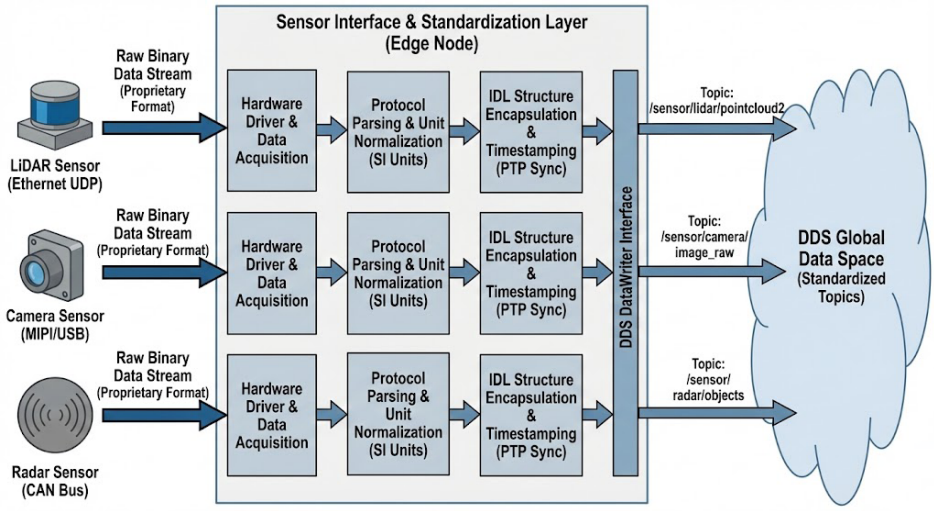


Fig. 5. Flowchart of standardized packaging and distribution of heterogeneous sensor data

3.2 Data Flow Protection Based on DDS Security Extension

High-altitude obstacle removal vehicles are specialized equipment used in high-risk missions, and the security of their information systems directly determines mission success and the safety of personnel and equipment. In a distributed, open network environment, sensing data is easily eavesdropped on, leading to leaks of the battlefield environment, while control commands can be easily maliciously altered to cause vehicle malfunctions. Traditional boundary defense strategies are ineffective for DDS networks with numerous nodes and dynamic interactions. Therefore, this research deeply integrates the DDS Security extension specification to create an intrinsic, fine-grained data flow protection system, ensuring the confidentiality, integrity, and availability of sensing and control information during transmission.

DDS Security adopts a plug-in architecture, allowing security mechanisms to be transparently enabled by configuring security policy files without modifying application-layer code. Its main protection system consists of three parts: authentication, access control, and encrypted transmission.

The authentication mechanism is based on Public Key Infrastructure (PKI). Before joining a DDS domain, each participant (LiDAR publishing node, path planning subscription node, etc.) must present an X.509 digital certificate issued by a trusted Certificate Authority (CA). The DDS security plugin uses a built-in handshake protocol for two-way authentication, allowing only nodes with valid certificates and not on the revocation list to establish a connection. This effectively prevents unauthorized joining by spoofed nodes.

Fine-grained access control policies. Following the principle of least privilege, digitally signed XML permission files precisely define the operations that each authenticated participant can perform on a given topic. For example, the perception module is authorized to “write” to the LiDARPointCloud topic but can only “read” the VehicleMotionState topic; the chassis control module is authorized to “read” various obstacle topics, but only it has permission to “write” to the critical ControlCommand topic. When a participant attempts to perform unauthorized publish or subscribe operations, the DDS middleware directly rejects them, logically isolating data streams with different security levels.

Flexible data encryption and integrity protection. DDS Security can configure different encryption strengths based on the sensitivity of the data stream. For sensitive topics containing core control commands or high-precision map data, a high-strength AES-GCM symmetric encryption algorithm is used to fully encrypt the data payload, along with a Message Authentication Code (MAC). Not only can it prevent third-party eavesdropping, but it can also detect data tampering and replay during transmission. For non-sensitive logs or status monitoring

data, only integrity verification or plaintext transmission can be performed to balance security overhead and system performance. The entire secure data stream processing is transparent to upper-layer applications. The DDS security plugin automatically encrypts and signs data before sending it, and automatically decrypts and verifies the signature after receiving it, thus ensuring that only legitimate subscribers can obtain the true plaintext data. The DDS Security data stream security protection mechanism is shown in Fig. 6.

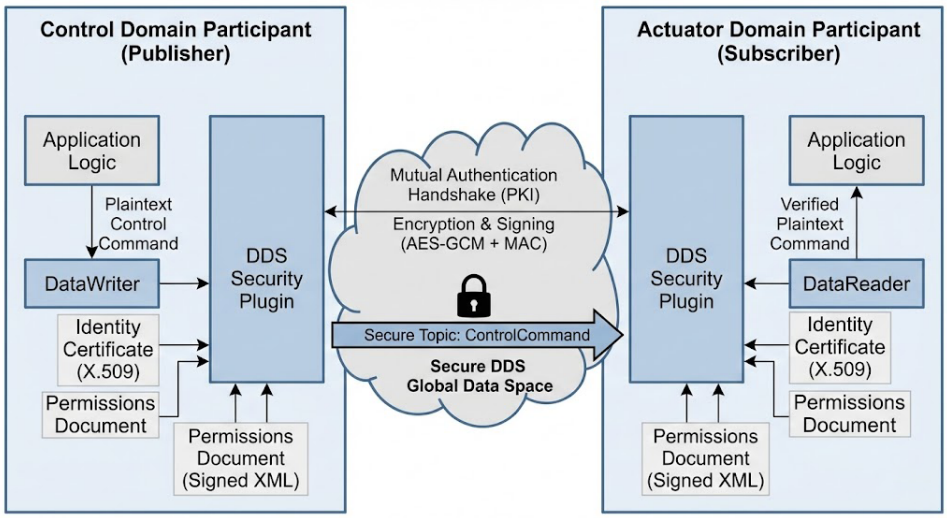


Fig. 6. DDS security data stream security protection mechanism diagram

3.3 Implementation of Fusion Sensing Service and Subscription Processing

After standardizing and publishing heterogeneous sensor data, the main task of multi-source sensing data stream processing shifts to the implementation of the fusion sensing service. This service plays a crucial role in the distributed architecture, collecting raw or pre-processed data streams from various sensors and providing them to the decision-making and planning module with a hierarchical and highly reliable unified environment model.

As an independent DDS domain participant node, the fusion sensing service operates in a closed-loop "subscribe-align-fusion-publish" manner. First, it subscribes to the standardized sensor topics defined in Section 3.1 by creating multiple DataReaders. Since different sensors have different sampling rates and transmission delays, the arrival time of data at the fusion node will inevitably differ. Therefore, the first step is to use timestamps to achieve time synchronization and spatial alignment. The service maintains a sliding time window, using interpolation or nearest neighbor matching to align different source data to the same time plane. Simultaneously, based on a pre-calibrated extrinsic parameter matrix, all sensor data is transformed to the vehicle reference coordinate system to eliminate spatial location differences.

After spatiotemporal alignment, the data is then fed into the core fusion algorithm module. A hybrid fusion architecture is used in the unstructured road environment of the plateau. For tracking dynamic obstacles, Extended Kalman Filter (EKF) is used, fusing radar target-level data and lidar clustering results for state estimation to obtain a stable list of targets with velocity and location information. For semantic understanding of complex terrain and static obstacles, a lightweight deep neural network model is used, fusing camera images and lidar point clouds at the feature level to obtain a 3D occupied grid map with semantic labels.

The final output of the fusion service is a highly abstract "environment model" topic, which includes the fused obstacle list, its own location information, and a local terrain map. This problem uses a DataWriter with a highly reliable, transient, and locally persistent QoS policy to publish to the global data space, ensuring that downstream key nodes can reliably obtain the latest environmental state.

On the subscription side, modules such as path planning and behavior decision-making act as consumers, subscribing to the FusedEnvironmentModel topic by creating corresponding DataReaders. The key processing on the subscription side is QoS policy matching and data validity verification. The DDS middleware automatically

checks whether the QoS of the publisher and subscriber matches; a connection is only established when the subscriber's request policy is not higher than the publisher's provided policy. Upon receiving the fused data, the subscriber first checks if the timestamp is within the allowable delay range, discarding outdated and invalid data. Then, based on its own task requirements, it transforms the global environment model data into a local planning coordinate system to generate a safe driving trajectory or execute obstacle avoidance actions. The publish-subscribe model based on DDS completely decouples the perception layer and the decision-making layer. The decision-making module does not need to know what sensor combinations are used at the underlying level, only the unified fusion result, greatly improving the modularity and scalability of the system. The internal implementation of the fused perception service and the interaction process with the subscriber are shown in Fig. 7.

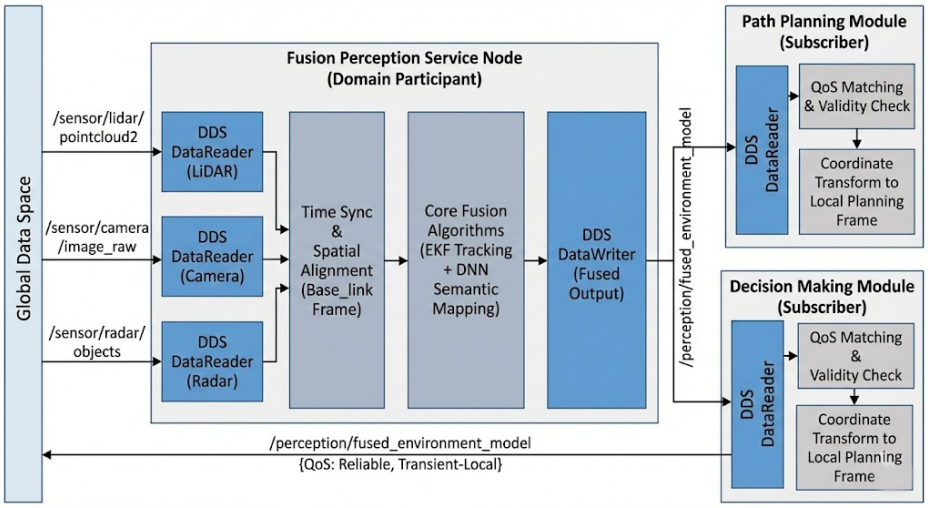


Fig. 7. Flowchart of the internal implementation of the fusion perception service and the interaction with the subscription client

4 Real-time Control Calculation and Computational Resource Scheduling

4.1 Software Componentization Implementation of Vehicle Dynamics Model

In a distributed software architecture, the perception layer provides environmental information, the decision layer generates the desired trajectory, and the control layer's task is to convert the upper-layer instructions into the control signals required by the chassis actuators, thereby ensuring the vehicle stably and accurately follows the target trajectory. The control algorithm for obstacle removal vehicles operating in high-altitude, low-adhesion, steep, and complex terrain requires real-time accuracy. This chapter mainly discusses how to transform complex vehicle dynamics models into reusable software modules, enabling efficient scheduling of real-time control tasks at edge computing resources.

The foundation for achieving precise control is an accurate mathematical description of the physical characteristics of the controlled object. Since high-altitude obstacle removal vehicles frequently travel on unstructured surfaces such as ice, snow, and gravel, tire lateral slip characteristics and road contact forces are nonlinear, making simple kinematic models insufficient for control requirements. Therefore, this paper establishes a seven-degree-of-freedom (7-DOF) nonlinear vehicle dynamics model, including longitudinal, lateral, yaw, and four-wheel rotational motions. To seamlessly embed the complex mathematical model into a DDS-based distributed architecture, a software componentization approach is used to encapsulate it as an independent standard software module. The key to componentization is interface standardization and black-

box internal implementation. The input and output interfaces of the dynamics model components are rigorously defined according to the interface definition language, making them participants in the DDS domain. The standardized input interface component uses a DDS data reader to subscribe to control command topics issued by the upper-level planning and control module. These topics include the desired front wheel steering angle and the driving or braking torque of each wheel. To accommodate testing of various control algorithms, the input interface can subscribe to virtual test excitation signals. The core solution engine component contains a fixed numerical integrator to perform the entire system simulation calculation. This solver solves the dynamic differential equations based on the current vehicle state vector and the received control input vector, using pre-calibrated vehicle physical parameters to obtain the vehicle state vector for the next time step. The vehicle state information obtained from the standardized output interface is encapsulated into DDS topics and then published to the global data space using a data writer. The output information includes the three-dimensional position, velocity, and acceleration of the vehicle's center of mass, as well as the vehicle's attitude angles and their velocities. This state data can be used not only by the underlying PID controller or model predictive controller but also as feedback information provided to the upper-level path planning module for replanning. By using a component-based approach, complex dynamic models are simplified into a virtual vehicle service with a standard DDS interface. This greatly improves the system's flexibility. During the simulation phase, it can be connected to a virtual environment for algorithm verification, and during the real-vehicle deployment phase, it can run as a state observer or prediction model on the onboard computing platform without requiring modifications to the code of other modules.

4.2 Dynamic Scheduling of Computing Resources Based on the Improved PSO Algorithm

The onboard computing platform of the plateau obstacle removal vehicle is a typical heterogeneous edge computing system, integrating various computing resources such as general-purpose processors, graphics processing units (GPUs), and field-programmable gate arrays (FPGAs). Different types of tasks have vastly different resource requirements. Deep learning inference tasks in environmental perception are computationally intensive and can be accelerated in parallel using GPUs. Sensor data preprocessing and feature extraction are often I/O intensive, which can be accomplished using FPGA pipelines. Decision planning and logic control are control intensive and are more suitable for execution on CPUs. The concurrency and load of various tasks vary greatly under different operating conditions. Traditional static resource allocation methods cannot adapt to dynamic changes, easily leading to some cores being overloaded while others are idle, thus affecting the real-time performance of the entire system.

Therefore, this paper proposes a dynamic resource scheduler using an improved particle swarm optimization (PSO) algorithm. This scheduler is an independent DDS service that periodically obtains the real-time load and temperature status of each computing core from the system monitor, and simultaneously retrieves the resource requirements of the tasks to be scheduled from the task queue, namely, estimated execution time, memory usage, and required hardware accelerator type. The core of the scheduling problem is to map tasks to be processed onto heterogeneous computing units to achieve multi-objective optimization, namely, maximizing system throughput, minimizing the completion time of high-priority tasks, and minimizing system energy consumption.

Standard PSO algorithms are prone to getting trapped in local optima for such discrete combinatorial optimization problems. An adaptive inertia weight strategy is used to improve standard PSO. In the early stages of iteration, particles are given a large inertia weight to enhance their global exploration ability. As iterations progress, the inertia weight is gradually reduced to improve the fine-grained search capability in local areas. The scheduler encodes each task mapping scheme into a particle, and the particle's fitness function is a comprehensive index of task-weighted completion time and system load balancing. After multiple iterations, the algorithm obtains the globally optimal mapping scheme, and the scheduler distributes tasks to the corresponding computing units for execution based on this scheme. This dynamic scheduling method ensures sufficient computing resources for control and solution tasks when resources are scarce or operating conditions change. The working principle of the dynamic computing resource scheduler is shown in Fig. 8.

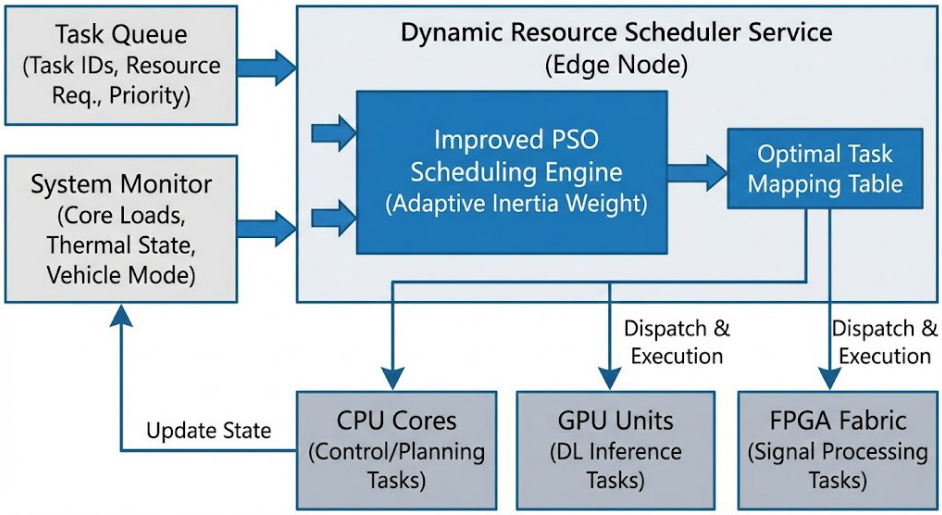


Fig. 8. Schematic diagram of the dynamic scheduler for computing resources

4.3 Real-time Distribution and Fault-tolerant Design of Control Commands

The critical chassis control commands generated by the control solution layer must be distributed to various actuator controllers with extremely high real-time performance and reliability. Due to the complex electromagnetic environment and severe mechanical vibrations at high altitudes, establishing a deterministic, low-latency, and fault-tolerant command distribution mechanism is the last line of defense for ensuring the safe operation of the entire vehicle.

This paper designs a dedicated real-time control command distribution channel based on the DDS middleware. First, a standardized 'ChassisControlCommand' topic is defined, whose IDL structure consists of three parts: the target control variable, the sequence number, and a precise timestamp. To meet the requirements of hard real-time, strict QoS policies are set on both the VCU publisher and actuator subscriber ends. Reliability is set to 'RELIABLE' to guarantee command arrival, the deadline is set to 10ms, consistent with the control cycle, forcing data updates within this window, and a high transmission priority is set to cope with network congestion.

From a fault-tolerant design perspective, an active safety system relying on heartbeat monitoring and multi-level fault response is created. The VCU periodically emits heartbeat signals using the DDS activity mechanism to declare its status. The DDS subscription client of the actuator node uses built-in QoS to implement two-level fault handling. When the 'on_deadline_missed' event is triggered, the actuator briefly maintains the valid instructions from the previous frame; when the 'on_liveliness_changed' event is triggered, it determines that the upper-level controller has failed, immediately initiates the local "safety failure" logic, and executes predefined actions such as emergency braking to ensure the vehicle remains in a safe state.

5 Results

System functional testing was conducted according to the principle of point-to-surface, step-by-step integration. First, independent functional tests were performed on basic software modules such as various sensor drivers, perception algorithm models, and control solvers. Then, integration testing was conducted, mainly verifying the interoperability implemented by various participants using the DDS middleware, ensuring that data publishing and subscription relationships could be correctly established and that cross-node data transmission could be achieved. Finally, system-level testing was performed, evaluating the closed-loop operation of the entire business process, including multi-source perception fusion, obstacle detection, autonomous obstacle avoidance planning, and chassis collaborative control, in a typical high-altitude obstacle removal operation environment. Verification shows that the designed distributed software architecture can meet the various operational requirements of the obstacle removal vehicle.

After successful functional verification, this paper conducted quantitative tests on the system's main performance indicators. The focus was on evaluating the system's real-time performance, throughput, resource utilization, and fault tolerance. Professional network analysis tools and performance monitoring software were used to measure the end-to-end transmission latency and jitter of key control commands under different network load conditions, thereby evaluating the effectiveness of the QoS strategy. Additionally, the concurrent throughput of the system when processing multiple high-resolution sensor data streams was examined. Furthermore, the usage of CPU, memory, and other resources by the DDS middleware and various application processes during operation was monitored to verify the effectiveness of the dynamic scheduling algorithm for computing resources. Finally, abnormal operating conditions such as artificially induced node failures and network interruptions were used to test the response speed and reliability of the system's fault detection and security degradation mechanisms. The comprehensive evaluation results demonstrate that the architecture exhibits excellent robustness under high-performance requirements.

6 Conclusions and Outlook

This paper addresses key technical challenges faced by high-altitude obstacle removal vehicles in intelligent upgrades under extreme geographical environments and complex operating conditions, including ensuring real-time communication, fusion of heterogeneous multi-source data, and edge computing resource scheduling. Starting with system architecture design, it studies and proposes a design scheme for a distributed software architecture based on DDS middleware. The main work and research conclusions of this paper are as follows:

This paper proposes a QoS-aware distributed DDS software architecture, breaking the traditional "siloed" system closedness of special vehicles. It decouples the perception, decision-making, and control functional domains, using a data-centric publish-subscribe approach to solve the problem of information silos between heterogeneous nodes. Especially for high-altitude operations with high real-time requirements, different QoS policy configuration schemes are designed, successfully establishing a logical communication channel with deterministic latency guarantees on an unreliable physical network.

To address the problem of limited computing resources at the vehicle's edge, a method for dynamic scheduling of heterogeneous resources using an improved particle swarm optimization algorithm is proposed. This mechanism can adjust the allocation of computing tasks among heterogeneous cores such as CPUs, GPUs, and FPGAs according to changes in vehicle operating conditions, thereby improving the overall throughput and speed of critical tasks, effectively solving the bottleneck caused by insufficient computing power during algorithm deployment.

A standardized encapsulation and deep fusion system for multi-source heterogeneous sensing data is established. Utilizing a unified data abstraction layer and spatiotemporal synchronization protocol, data from various sensors such as LiDAR and vision cameras is aggregated, and deep learning algorithms are used to improve the system's environmental perception robustness, enabling it to function well even in harsh weather conditions such as strong light and snow.

A comprehensive security system combining active defense and passive fault tolerance is established. The communication layer incorporates a DDS security extension mechanism to ensure the confidentiality and integrity of data streams, while the control layer uses heartbeat detection and multi-level fault response logic to ensure the inherent safety of the vehicle in the event of communication anomalies or node failures.

Although this paper has achieved some results in the distributed software architecture design of plateau obstacle removal vehicles, with the continuous development of unmanned combat requirements and the rapid advancement of artificial intelligence technology, there is still significant room for further research. Future research directions include: From single-vehicle intelligence to cluster collaboration. The current architecture primarily focuses on the intelligent transformation of individual vehicles. Future research will continue on cross-vehicle data sharing and distributed consensus algorithms based on DDS (Data Streaming System), exploring methods for collaborative formation, task allocation, and cooperative operation of multiple obstacle removal vehicles in environments without a central network. This will address larger and more complex battlefield engineering support tasks.

A ubiquitous "cloud-edge-device" interconnected architecture will be created. Based on research into low-Earth orbit satellite internet and 5G/6G mobile communication technologies, research will be conducted on technologies that seamlessly integrate in-vehicle DDS real-time networks and wide-area command and control networks. A cloud-based digital twin system will be created to manage the health of vehicles throughout their

entire lifecycle in high-altitude environments, enabling remote takeover and continuous online algorithm updates (OTA).

The application of reinforcement learning relying on virtual-to-real (Sim-to-Real) transfer will be strengthened. To address the difficulty of data acquisition in extreme high-altitude environments, a high-fidelity virtual simulation training environment will be created. Optimal control strategies will be trained through extensive trial and error in simulations. Research will be conducted on how to seamlessly transfer these strategies to the real vehicle controller, thereby improving the autonomy and environmental adaptability of obstacle removal operations.

7 Acknowledgement

The research of the authors was supported by Industrial Internet innovation and development project of MIIT, China.

References

[1] P. Merino, G. Mujica, J. Señor, J. Portilla, A modular IoT hardware platform for distributed and secured extreme edge computing, Electronics 9(3)(2020) 538.
https://doi.org/10.3390/electronics9030538

[2] T. Magedanz, F. Schreiner, QoS-aware multi-cloud brokering for NGN services: tangible benefits of elastic resource allocation mechanisms, in: Proc. 2014 IEEE Fifth International Conference on Communications and Electronics (ICCE), 2014.
https://doi.org/10.1109/CCE.2014.6916698

[3] J. Konrad, N. Bozinovic, Interpretation of uniform translational image motion: DCT versus FT, in: Proc. 2002 International Conference on Image Processing, 2002.
https://doi.org/10.1109/ICIP.2002.1039942

[4] P. Gil-Jimenez, H. Gomez-Moreno, P. Siegmann, S. Lafuente-Arroyo, S. Maldonado-Bascon, Traffic sign shape classification based on Support Vector Machines and the FFT of the signature of blobs, in: Proc. 2007 IEEE Intelligent Vehicles Symposium, 2007.
https://doi.org/10.1109/IVS.2007.4290143

[5] X. Li, Z. Liu, Z. Huang, Attention-based radar PRI modulation recognition with recurrent neural networks, IEEE Access 8(2020) 57426-57436.
https://doi.org/10.1109/ACCESS.2020.2982654

[6] S. Ramasamy, V. Chandrasekar, A.M. Viswa Bharathy, Classification of nutrient deficiencies in plants using recurrent neural network, in: Proc. 2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC), 2022.
https://doi.org/10.1109/ICAECC54045.2022.9716641

[7] Z.-T. Ye, P.-J. Wu, H.-C. Kuo, Feasibility study of spectral detection bovine serum albumin, in: Proc. 2021 IEEE 6th Optoelectronics Global Conference (OGC), 2021.
https://doi.org/10.1109/OGC52961.2021.9654321

[8] Z. Wang, J. Chen, B. Wang, Dynamic clustering recommendation algorithm for two-layer graph attention network, in: Proc. 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC), Data Science and Computational Intelligence (DSCI), and Smart Computing, Networking and Services (SmartCNS), 2019.
https://doi.org/10.1109/IUCC/DSCI/SmartCNS.2019.00098

[9] N. Bekiaris-Liberis, R. Vazquez, Nonlinear bilateral full-state feedback trajectory tracking for a class of viscous Hamilton-Jacobi PDEs, in: Proc. 2018 IEEE Conference on Decision and Control (CDC), 2018.
https://doi.org/10.1109/CDC.2018.8619363

[10] R. Patelski, D. Pazderski, Parameter identifying disturbance rejection control with asymptotic error convergence, IEEE Robotics and Automation Letters 9(2)(2024) 1035-1042.
https://doi.org/10.1109/LRA.2023.3339942

[11] S. Zheng, Y. Dong, S. Yang, J. Liang, Research on AI electronic fuel injection of agricultural hybrid UAV, in: Proc. 2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2022.
https://doi.org/10.1109/IMCEC55388.2022.10020051

[12] J. Zhao, M. Yang, Y. Zhao, X. Hu, W. Zhou, H. Li, MCMARL: parameterizing value function via mixture of categorical distributions for multi-agent reinforcement learning, IEEE Transactions on Games 16(3)(2024) 556-565.
https://doi.org/10.1109/TG.2023.3310150

[13] H. Jiang, E. Du, N. Zhang, Z. Zhuo, P. Wang, Z. Wang, Y. Zhang, Renewable electric energy system planning considering seasonal electricity imbalance risk, IEEE Transactions on Power Systems 38(6)(2023) 5432-5444. https://doi.org/10.1109/TPWRS.2022.3229568

[14] Y. Zhang, X. Jiang, Z. Lin, A general k-anycast DTN routing, in: Proc. 2010 International Conference on Mechanic Automation and Control Engineering, 2010. https://doi.org/10.1109/MACE.2010.5536827

[15] G.K. Michelon, W.K.G. Assunção, P. Grünbacher, A. Egyed, Analysis and propagation of feature revisions in preprocessor-based software product lines, in: Proc. 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2023. https://doi.org/10.1109/SANER56733.2023.00035

[16] K.A. Zulkepli, H. Zamzuri, M.A.A. Rahman, W.J. Yahya, M.A.B. Zakaria, M.Z. Azmi, F.R.A. Zakuan, N.H.F. Ismail, M.S. Zakariya, I-Drive: modular system architecture and hardware configuration for an intelligent vehicle research platform, ARPN Journal of Engineering and Applied Sciences 12(14)(2017) 4259-4264. https://eprints.utm.my/75381/

[17] D. Comaneci, C. Dobre, Securing networks using SDN and machine learning, in: Proc. 2018 IEEE International Conference on Computational Science and Engineering (CSE), 2018. https://doi.org/10.1109/CSE.2018.00034

[18] S.S. Kirubakaran, V. Ebenezer, P. Santhiya, G. Manojkumar, S. Sophia, J.A.S. Snowlin Preethi, An effective study on different levels of honeypot with applications and design of real time honeypot, in: Proc. 2023 2nd International Conference on Edge Computing and Applications (ICECAA), 2023. https://doi.org/10.1109/ICECAA58104.2023.10212345

[19] S. Yuan, J. Ye, Energy consumption and fault analysis of data center power system, in: Proc. 2023 International Conference on Data Science and Network Security (ICDSNS), 2023. https://doi.org/10.1109/ICDSNS58469.2023.10245678

[20] F. Liu, Y. Shuai, X. Zhang, Y. Xiong, Y. Zeng, G-computation demonstration in estimating causal effects with time-dependent confounding, in: Proc. 2022 IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI), 2022. https://doi.org/10.1109/SEAI55746.2022.9832345

[21] CSDN Blog, Discussing Hardware Platformization - Hardware Modularization. <https://blog.csdn.net/qq_43131421/article/details/138850310>, 2024 (accessed 14.07.26).

[22] Sina Technology, Passenger-type high-altitude boosted vehicle. <http://finance.sina.com.cn/tech/digi/2025-03-30/doc-inermezq9361796.shtml>, 2025 (accessed 14.07.26).

[23] J. Sitthiworachart, Flipped classroom approach for the development of thinking skills and the fostering of student engagement, in: Proc. 2023 11th International Conference on Information and Education Technology (ICIET), 2023. https://doi.org/10.1109/ICIET56899.2023.10111223