Education Reform and Development, 2025, Volume 7, Issue 8
https://ojs.bbwpublisher.com/index.php/ERD
Online ISSN: 2652-5372
Print ISSN: 2652-5364

Design and Implementation of An Intelligent Robot Control System Based on Java Inheritance and Polymorphism Characteristics

Dan Liu*

College of Advanced Technology, Jiaotong Road Campus, Shanghai Business School, Shanghai 200065, China

**Author to whom correspondence should be addressed.*

Abstract: With the rapid development of robotics technology, designing an efficient, flexible, and scalable control system has become the key to improving robot performance. This paper proposes a design and implementation scheme for an intelligent robot control system based on Java inheritance and polymorphism characteristics. By fully utilizing the object-oriented features of the Java language, particularly its inheritance and polymorphism mechanisms, a modular control system architecture is constructed. This architecture not only improves code reusability and maintainability but also enhances system flexibility and scalability. In the implementation process, key aspects such as sensor data processing, control algorithm design, and actuator control strategies are thoroughly explored. Comprehensive system testing verifies the effectiveness of the system. The research results indicate that intelligent robot control systems based on Java inheritance and polymorphism exhibit excellent performance, functionality, and stability, providing new ideas and methods for the development of intelligent robot technology.

Keywords: Java inheritance; Polymorphic characteristics; Intelligent robots; Control system; Sensor data processing; Control algorithm design; Actuator control strategy

Online publication: September 4, 2025

1. Analysis of research status

With the rapid advancement of science and technology, robotics has become an indispensable part of modern industrial production. From precise operations on automated production lines to efficient management of intelligent warehousing systems, and humanized interactions in the service field, robotics is gradually changing our working methods and lifestyle patterns. In this transformation, the Java programming language has shown unique advantages in developing complex robot control systems due to its cross-platform compatibility, object-oriented design philosophy, and advanced security features. Java's inheritance mechanism allows developers to create hierarchical class structures, achieving code reuse and function expansion through the relationship between parent classes and subclasses. Polymorphism is another major feature of Java, which allows us to use variables

of parent class types to reference subclass objects and execute corresponding methods according to the actual object type at runtime. This feature brings great flexibility to the design of robot control systems. Designing and implementing intelligent robot control systems based on Java's inheritance and polymorphism features can not only improve code reusability and scalability but also reduce system maintenance costs, promoting the wider application of robotics in modern industrial production [1].

1.1. Current challenges in research

Research on intelligent robot control systems based on Java's inheritance and polymorphism features has made certain progress. Many researchers have fully utilized the object-oriented features of the Java language, especially its inheritance and polymorphism functions, to design modular and scalable architectures for intelligent robot control systems. In existing studies, some researchers have conducted three-dimensional simulations of humanoid robots through Java 3D technology, verifying the effectiveness of Java in the design of robot control systems. In addition, some researchers have built a motion simulation system for space robots based on Java 3D to evaluate and optimize the robot's motion performance. In practical applications, Java is also used to build stable and reliable communication mechanisms and complex control algorithms, such as the communication and control system of the new ARV underwater robot. Despite certain achievements, how to further improve the real-time performance, stability, and security of the system, and how to realize more intelligent and autonomous control strategies, remains a challenge in current research [2].

1.2. Research objectives and innovations

This paper is committed to designing and implementing an intelligent robot control system based on Java's inheritance and polymorphism features. It leverages the powerful functions of the Java language, especially its inheritance and polymorphism features, to build a flexible, scalable, and easy-to-maintain robot control system. The innovations of this paper are mainly reflected in the following aspects:

(1) Hierarchical Class Structure: By in-depth analysis of Java's inheritance mechanism, a hierarchical class structure is designed to improve code reusability and realize system modularization.

(2) Application of Polymorphism: Making full use of Java's polymorphism feature, a series of interfaces and abstract classes are defined to realize unified management and control of different types of robot equipment, increasing code flexibility and scalability.

(3) Efficient Data Processing and Control Algorithms: Proposing Java-based sensor data processing and actuator control strategies to ensure that robots can respond to changes in the external environment accurately and quickly [3].

2. Java inheritance and polymorphism features

2.1. Java inheritance mechanism

Java's inheritance mechanism is one of the core features of object-oriented programming. It allows the creation of a hierarchical class structure, where subclasses can inherit the properties and methods of parent classes. This mechanism not only improves code reusability but also helps build more modular and maintainable software systems. In Java, class inheritance is implemented through the extends keyword. Subclasses can inherit all non-private properties and methods of parent classes and can also add their own unique properties and methods. Java also supports multi-level inheritance, meaning a subclass can inherit from another subclass, forming an

inheritance chain of classes. In intelligent robot control systems, Java's inheritance mechanism can be used to design different types of robot controllers. For example, a general RobotController class can be created as the parent class, containing the common properties and methods of all robot controllers. Then, for different types of robots (such as wheeled robots, humanoid robots, etc.), corresponding subclasses can be created to inherit from the RobotController class and add specific control logic and algorithms [4].

2.2. Manifestation of polymorphism in Java

Polymorphism is one of the core concepts of object-oriented programming, which enables different objects to produce different behavioral responses to the same message. In Java, polymorphism is mainly achieved through method overloading and overriding. Method overloading occurs within the same class, allowing the definition of multiple methods with the same name but different parameter lists. Method overriding occurs between parent and child classes, where subclasses can redefine methods already present in the parent class to achieve specific behaviors. In intelligent robot control systems, polymorphism can be widely applied to process different types of sensor data and actuator control commands. For example, define a general Sensor interface and multiple specific classes that implement this interface (such as TemperatureSensor, PressureSensor, etc.). These specific classes can override the methods in the Sensor interface to provide specific data collection and processing logic. In the control system, different types of sensor objects can be created according to actual needs, and their methods can be called to obtain sensor data. Due to Java's polymorphism, these calls will be automatically bound to the correct implementation, thereby simplifying the code structure and improving the flexibility of the system [5].

2.3. Application of Java inheritance and polymorphism in control systems

In intelligent robot control systems, Java's inheritance and polymorphism features are crucial. Inheritance makes the class hierarchy clear, facilitating the construction of modular control systems. For example, by inheriting from the "Robot" base class, subclasses such as "service-type" and "industrial-type" robots can be derived. Polymorphism improves system flexibility, allowing the use of a unified interface to handle different objects. Defining general interfaces such as "sensor data" and "actuator control," implemented by various types, enhances the scalability and maintainability of the system [6].

3. Overall design of the control system

3.1. System architecture design

The intelligent robot control system adopts a layered architecture, divided into four layers: data acquisition, processing, decision-making, and execution. The data acquisition layer ensures accurate and real-time sensor data; the processing layer extracts useful information; the decision-making layer formulates behavior strategies; and the execution layer drives the robot's actions. The layered design achieves modularization, improving reusability and testability [7].

3.2. Division of system function modules

The system is divided into four modules: sensor, data processing, control, and execution, based on modularity, scalability, and maintainability. The sensor module collects environmental information; the data processing module preprocesses data; the control module generates instructions; and the execution module drives actions. A reasonable division facilitates independent development, testing, and functional expansion [8].

3.3. System interaction design

The system interaction design is based on message passing, building an efficient and scalable network. Message queues balance processing speeds to prevent congestion and loss. Through confirmation mechanisms, timestamps, and retransmissions, reliable and real-time messaging is ensured. Java message middleware is used to achieve decoupled asynchronous communication, improving performance and scalability [9].

4. Implementation of the control system

4.1. Design of control algorithms

The PID algorithm calculates deviations and outputs control quantities through proportional, integral, and differential adjustments to achieve precise control. In the intelligent robot system, PID controllers are designed for position and speed. To enhance flexibility and scalability, the PID algorithm is encapsulated into an independent class, providing interfaces for invocation [10].

4.2. Sensor data processing

The sensor data processing module receives raw data and performs filtering (mean, median, Kalman, etc.) and feature extraction (signal processing and machine learning). To improve scalability and maintainability, data processing is encapsulated into an independent class with clearly defined interfaces. Multi-threading and asynchronous processing are adopted to enhance the system's response speed and anti-interference capability [11].

4.3. Actuator control strategy

The actuator control adopts the PWM strategy, which controls the operation by adjusting the pulse width to realize continuous and smooth power regulation, meeting the requirements of real-time precision. The PWM control is encapsulated into an independent class, providing interfaces for invocation, and has been tested and optimized to ensure stability and reliability [12].

5. System testing and optimization

5.1. Design of testing scheme

The testing scheme covers functional, performance, and stability tests. Combining black-box and white-box testing, it verifies module functions, evaluates system response and processing capabilities, and inspects fault-tolerance and recovery mechanisms to improve testing efficiency and accuracy [13].

5.2. Analysis of test results

After comprehensive testing, all modules and functional points of the system performed normally without obvious functional defects or abnormalities. In terms of performance testing, the system could still maintain a high response speed and stability when processing large amounts of data and executing complex algorithms. In the stability testing phase, the system could operate stably under various adverse conditions. Meanwhile, some problems and deficiencies were found, such as performance fluctuations under specific conditions, and the need to improve the accuracy and efficiency of sensor data processing. Corresponding improvement suggestions and optimization measures were put forward for these problems [14].

5.3. System optimization strategies

The system performed normally after testing, with stable performance and fast response speed. However, there are problems such as performance fluctuations and the need to improve sensor data processing. Improvement suggestions and optimization measures have been proposed [15].

6. Conclusion

This paper discusses the design of an intelligent robot control system based on Java inheritance and polymorphism, constructing a modular, extensible, and flexible framework to improve maintainability and scalability. The adaptability of the system is enhanced through inheritance and polymorphism, and advanced technologies and algorithms improve performance and stability. System testing verifies its effectiveness, and optimization suggestions are put forward. The research results provide new ideas for intelligent robot control and have positive significance for future technological development. Future research can explore system integration, autonomous navigation, intelligent perception, etc., focus on safety and privacy protection, and develop more advanced, intelligent, and safe robot control systems.

Funding

"National Association of Computer Basic Education in Colleges and Universities" (Project No.: 2025-AFCEC-568); "Shanghai Key Laboratory of Computer Software Testing and Evaluation" (Project No.: SSTL_YJ_202501)

Disclosure statement

The author declares no conflict of interest.

References

[1] Chen Y, Li Y, 2025, Design and Implementation of Web-Based Employee Management System. Technology Innovation and Application, 15(16): 129–132.

[2] Zou Y, Shan Z, Sun D, et al., 2025, Design of Data Processing and Application Forwarding System Based on Object Storage. Computer Engineering and Design, 46(2): 447–456.

[3] Guo Z, Hou P, Yang M, 2025, Design and Implementation of Online Optical Diffraction Demonstration System Based on Java and Matlab. Journal of Gansu Normal Colleges, 30(1): 70–74.

[4] Wang Y, 2013, Implementation of Communication and Control System for New ARV Underwater Robot Based on JAVA. 2013 Conference Article.

[5] Wu Y, Li X, 2025, Design of Vegetable Sales System Based on Java Language. Technology Innovation and Productivity, 46(2): 130–133.

[6] Zhang F, Liu D, Qiu L, Zhao W, 2020, Construction of Teaching Resources for "Java Programming" Oriented to Software Development Practical Ability. Software Engineering, 23(4): 60–62.

[7] Wang X, Feng F, 2025, Design of "Internet of Things Technology" Course Learning System Based on Java Technology. Internet of Things Technology, 15(1): 153–157.

[8] Li X, Feng W, 2020, Research and Practice on School-Enterprise Cooperation in Joint Development of "Java

Project Actual Combat" Course. Computer Products and Circulation, 2020(5): 194.

[9] He D, 2020, Research on Android Software Development Based on Java Language. Telecom World, 27(4): 62–63.

[10] Gao H, 2020, Preliminary Exploration of JAVA Programming Application Based on Computer Software Development. Telecom World, 27(4): 119–120.

[11] Yang L, 2020, Discussion on Characteristics and Technology of Java Programming in Computer Software. Computer Programming Skills & Maintenance, 2020(4): 47–49.

[12] Yuan L, 2020, Application of Java Language in Computer Software Development. Network Security Technology and Application, 2020(4): 79–80.

[13] Li X, Liu W, Ding C, 2020, Teaching Practice of Java Programming Course for Multi-Level Ability Cultivation. Computer Era, 2020(4): 81–84.

[14] Zhang T, Fu J, 2020, Application of Balance and Rebalance Learning Theory in Java Programming Teaching. China Information Technology Education, 2020(8): 102–104.

[15] Wang W, Yao Y, 2020, Online Java Editor Based on Java Web. Electronics World, 2020(7): 76–77.