Design and Implementation of a PLC-based Multi-Axis server Control system

Xu Ji[1†], Yifei Guo[1*†], Ying Zhou[1†]
{493591532@qq.com, 3092307213@qq.com, 2806654757@qq.com}
[1]Chengdu Technological University, No. 1, Section 2, Zhongxin Avenue, Chengdu, 611730, China
∗ Corresponding author.
† These authors contributed equally.

Abstract. In modern industry, the demand for motion control systems is increasing and there is an urgent need for stable electrical systems and a high-performance automatic control system to meet the needs of complex applications. To this end, this study conducted an in-depth investigation into a control scheme based on a programmable logic controller (PLC) and multi-axis servo units. By integrating the requirements of modern industrial motion control systems, a multi-axis servo control system based on Mitsubishi's Q-series PLC was designed and implemented. With the rapid development of AI technology, large language models can now perform some of the computational functions of PLCs. Therefore, the system combines PLC control technology, multi-axis servo control technology, and large language modeling to construct a high-performance servo control system that achieves precise positioning and multi-axis linkage, meeting industrial demand for high-performance servo systems.

Keywords: PLC, Large Language Model (LLM), Multi-axis linkage, Servo positioning

1 Introduction

With the continuous improvement of industrial automation level and the pursuit of high precision and efficiency in production processes. In modern industrial production, multi-axis servo systems are widely used in industrial control fields such as CNC machine tools, robots, and industrial automation. With the rise of Industry 4.0 and intelligent manufacturing, the performance requirements for multi-axis servo control systems are becoming increasingly high. They not only require high precision, high speed, and high stability [1], but also need to have functions such as intelligence, remote monitoring, and fault prediction.

In this context, a multi-axis servo control system based on PLC has emerged. By combining PLC with servo motors, the system can achieve precise control of multiple servo motors, meeting the needs of various complex production processes. With the continuous development of AI technology, cloud computing, big data, and other technologies, multi-axis servo control systems based on PLC

IIKI 2025, December 18-21, Chengdu, People's Republic of China

DOI 10.4108/eai.18-12-2025.2365261

are also facing opportunities and challenges. How to combine these new technologies with multi-axis servo control systems to achieve more intelligent, remote, and automated control is also one of the key research directions at present.

With the advancement of technology, PLC-based multi-axis servo control systems continue to develop and improve [2].Driven by the concepts of green manufacturing and sustainable development, PLC-based multi-axis servo control systems are beginning to develop towards intelligence, energy efficiency, and environmental friendliness. The main development directions are as follows: networking and intelligence [3]; High performance and precision [4]; Modularization and integration [5].

Therefore, this project is based on multi-axis servo motion control and utilizes open-source AI models for decision-making. Combining the Gomoku game with a multi-axis servo control system to achieve intelligent and diversified servo motion control, showcasing multi-axis servo control in the form of a game. This application not only achieves complex motion control tasks but also confirms the stability system in industrial applications, and opens up new ideas and possibilities for its application in more fields.

2 System Design

This system adopts the control scheme of "Gobang algorithm module+motion control system" [6], and constructs a multi-axis servo control system that can realize complex motion control through the SSCNet fieldbus network of the device layer and the TCP/IP network double-layer network architecture of the control layer. Through the combination of software and hardware, we have successfully built a gobang man-machine combat system that combines interest and technology. The system is mainly composed of the following parts, as shown in Figure 1.

2.1 Gobang algorithm module

Game bottom module: as the base of the whole algorithm, the game bottom module determines the data format used by the motion system. If you want to completely present the game match process in the human-computer interaction interface, the design of the bottom data is also particularly important.

Local large language model: as the decision-making body of the whole system, the large language model connects the PLC Ethernet module with the local PC, creates a TCP server by using the socket library in Python, so that the large language model can receive data, and then through a series of internal calculations, gets the point with the highest score of the current performer (i.e. the best bit), and finally converts the chessboard data containing this point into a two-dimensional array format that can be recognized by PLC, and sends it back to the upper-end computer, and then sent back to PLC by the upper computer. If the returned data does not conform to the identification format of PLC, it may lead to system failure or hardware risk. Therefore, the data format must be detected before transferring the data back to the PLC.

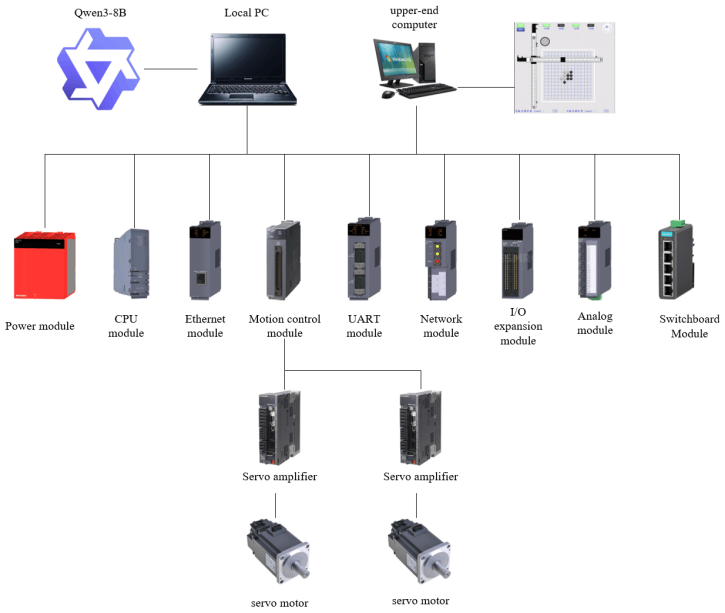

Fig. 1. system Components.

2.2 Motion control system

Upper-end computer: as the human-computer interaction interface and command center of the system, it monitors the system status in real time through powerful computing ability and visualization function, issues control instructions, and analyzes and sorts out the data.

PLC control system: PLC is the core hub of the whole system. It is responsible for receiving the instructions from the upper-end computer, converting them into specific control actions, and collecting various signals on site to feed back to the upper-end computer. With its stable and reliable performance, PLC ensures the accurate and efficient control of the system.

Multi-axis servo system: as the key link connecting the motion module and the servo motor, the servo amplifier can amplify and adjust the control signal output by the PLC to meet the driving requirements of the servo motor and ensure the accurate operation and rapid response of the motor. The servo motor is the actuator. By cooperating with the servo amplifier, the motion state is adjusted according to the control command to realize the stable control of the controlled object.

3 Gobang Algorithm Module Design

Gobang game algorithm mainly includes the following steps: initialization of chessboard, chess pieces, manipulator,and other data; Game mode selection, such as PvP and PVE;Analysis of man-machine drop points, find the best drop point through the large language model;Action execution

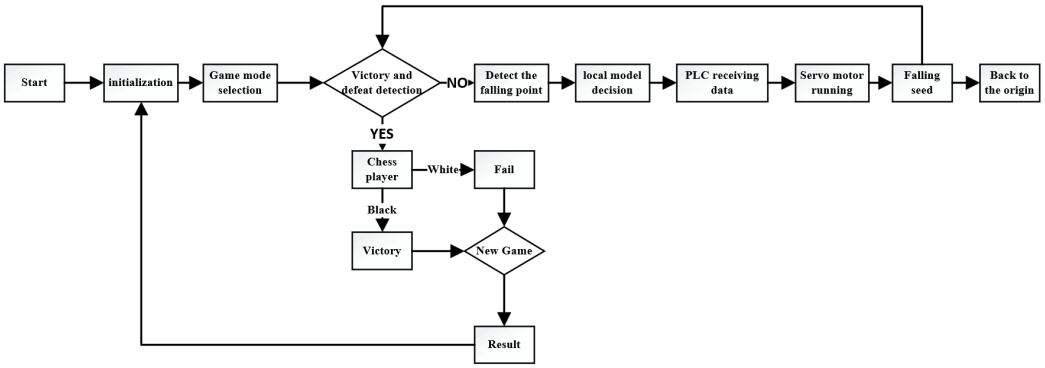

Fig. 2. Gobang algorithm flow chart

and origin return; Judge the victory or defeat. If it is not over, you need to switch the wheel. The flow chart of the game algorithm is shown in Figure 2.

According to the algorithm flow chart, the algorithm is mainly divided into the following modules for design:

3.1 Game bottom module design

Chessboard: the chessboard of Gobang is usually a 15*15 square. When performing the drop operation, you only need to click the corresponding empty point on the upper-end computer. In order to ensure that each drop point of the chessboard has corresponding data, a total of 15*15, or 225 data points, is required. We can create a two-dimensional array (0..14, 0..14) of word type with a length of 15 in PLC, which is named "button" and represented as a button in PLC. Each array element is represented by a word-type variable. The array range is set to 0-14, representing 1-15 rows and 1-15 columns on the chessboard.

Chess pieces: in this array, each element uses an integer to represent the state of the chess pieces; 0 represents the empty point, 1 represents the white, and 2 represents the sunspot.The chessboard state can be accurately reflected in the array.

Drop: to complete the drop in the upper-end computer,you first need to scan the chessboard circularly to determine whether a point is pressed. If a point is pressed after scanning,record the coordinates of the point and assign a value to the point according to the color of the wheel.

Victory detection and game saving: when a game is played,a victory detection will be performed after each move. When the number of consecutive beads is greater than or equal to five, the game will end, and the winner will be displayed according to the current round.

3.2 Large language model module

a) Advantages of a large language model: The rapid development of large language models in recent years it provided a new way to deal with complex tasks. The mainstream large language

model uses a transformer architecture, a self-attention mechanism[7], and self-supervised learning training to complete the Gobang game.

Capture the global association of chess pieces' positions: each position of Gobang is not isolated, and the self-attention mechanism can directly model the dependency between any two positions by calculating the association weight between each position and all other positions.

Pay attention to the key areas of the chessboard: the winning method of Gobang is "five in a row", and the attention scores of positions and key points that can form "live three", "rush four" will be significantly improved.

Multi-angle analysis of the situation: the multi-head self-attention can analyze the chessboard situation from different aspects through multiple parallel "attention heads", and pay attention to multiple enemy and our local ligatures to achieve attack-defense balance.

Reduce training costs: self-supervised learning can directly use the original chessboard data without manual annotation and quickly accumulate data experience.

Strengthen learning ability: The large language model can independently master the core rules of Gobang from the data and lay the foundation for subsequent decisions.

b) Construction of prompts: In order to make the large language model understand the task objectives and output the expected results, the construction of prompts can be used as a bridge between the large language model and the input data, so as to promote the generation of the large language model [8]. The construction of prompts can be divided into the following five steps.

Clear objectives: before constructing the prompt words, it is necessary to clarify the specific tasks faced by the model. In this task, the specific task faced by the large language model is to find the best landing point through the input two-dimensional array of Gobang, so the hint word constructed in this part can be "find the best landing point through the input two-dimensional array of chessboard information, and output it in the form of a two-dimensional array".

Understand model capabilities: Different models may exhibit varying performances when handling distinct types of tasks. Understanding the characteristics of these models can facilitate the construction of more effective prompts.

Initial design prompt: The initial prompt serves to provide context information, and should be as clear and specific as possible to help the model understand the task. In this task, the initial prompt can be designed as "you are an AI that focuses on playing Gobang and can be analyzed through a two-dimensional array of checkerboards".

Testing and iteration: Use the designed prompt words to interact with the model and observe the output of the model. If the output of the model does not meet the expectation, you need to adjust the prompt words according to the possible reasons.

Evaluation and optimization: If the prompts perform well, the cue words can be tested on different data sets and further optimized according to the results.

To sum up, the prompt in this task can be set as: "You are an AI who focuses on playing Gobang. Please analyze and find the best landing point according to the input two-bit array chessboard information, and output it in the form of a two-dimensional array."

c) Construction of dataset: To construct a high-quality fine-tuning dataset for a large Gobang model, the key lies in balancing data reliability, sample effectiveness, and model adaptability. By downloading publicly available, high-quality professional game records in the standard SGF chess

format and using them as source files, we can build a standardized dataset. The specific considerations are as follows:

Complete core information: the data set must contain coordinates, steps, the outcome of the game, and the chess player. Deduplication and filtering: Filter out short matches (fewer than 5 steps, lacking tactical value) and long matches (more than 200 steps, prone to exceeding the model's context limit).

File Format: The JSONL format is utilized, with core fields following the standard instruction+input+output structure. The parameters must conform to a two-dimensional array format, necessitating the creation of a script to establish a one-to-one correspondence between A-O and 0-14.

Introduce the scoring mechanism of placement: score the number of consecutive pieces of the same color and the number of blocked pieces of different colors around each placement according to priority, and extract the placement with the highest score and convert it to JSONL format.

Division Rationality: training set: verification set=8:2, ensuring that the verification set covers different stages of the game and tactical types.

d) Parameter-efficient fine-tuning (peft) [9]: The format output from the PLC to the large language model is a two-dimensional array. Without training the model, the model may appear: the meaning of "0", "1", "2" in the two-dimensional array cannot be correctly interpreted; Do not understand the rules and legitimacy of Gobang; Have no strategy; The output format is unstable and cannot be parsed by PLC; Unable to handle complex situations and other issues.

To solve this problem, we need to fine-tune a pre-training language model (such as qwen3-8b) and update the pre-training parameters. The formula is as follows:

$$W_0 + \triangle W \tag{1}$$

W_0 represents the initialization parameters of the pre-training model, and represents the parameters that need to be updated. If it is a full parameter fine-tuning, the parameter quantity $\triangle W \approx W_0$ is very expensive.

Peft is the core technology in the field of large language model fine-tuning. Its purpose is to quickly adapt to downstream tasks while solving the defects of full-scale fine-tuning of large models and retaining the general ability of model pre-training. Its advantages are as follows.

High efficiency and energy saving: fewer model parameters are required for fine-tuning, and only 0.1%-10% of model parameters are required to be fine-tuned, which greatly reduces the demand for computing power and can be supported by an ordinary GPU.

Avoid forgetting: full fine-tuning can easily lead to the model forgetting the pre-training knowledge. Peft only modifies local parameters to better retain the original ability.

Among many peft methods, low rank adaptation (LORA) is a popular method with excellent performance. This technology will be used in this task. For Lora, only fine-tuning is required. Assuming that the pre-training matrix is $W_0 \in \mathbb{R}^{d\times k}$, its update can be expressed as

$$W_0 + \triangle W = W_0 + BA, B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k} \tag{2}$$

During Lora training, W0 is fixed, and only the training parameters are. In the reasoning process, Lora almost does not introduce additional influence latency. The number of parameters is

significantly reduced compared with full parameter tuning, and the performance is better than other efficient parameter tuning methods, which is basically the same as or even higher than full parameter tuning. Therefore, Lora is selected to fine-tune the model [10].

e) Determination of fine-tuning parameters: To determine the parameters of the large language model fine-tuning, it is necessary to combine the model type, task scenario, data size, and other factors to judge. In the training process, it is necessary to balance the training effect, efficiency, and generalization ability. If the parameter setting is not appropriate, the model may have overfitting or underfitting, resulting in the generation of results that do not meet expectations.

For Gobang tasks, we can set fine-tuning parameters as follows.

Learning rate: it is determined by observing the decline of training loss. If the training loss fluctuates greatly, it indicates that the learning rate is too high and needs to be reduced. If the training loss decreases too slowly, the learning rate is too low and needs to be improved. The learning rate can be set to 3e-5 to observe the training effect.

Epochs: Observe the decline of verification loss. If verification loss continues to decline, you can increase the number of training rounds. If verification loss does not fall but rises, you need to reduce the number of training rounds. According to the data set size, such as the data set of 4000 samples, the epochs can be set to 8.

Weight decay: Weight attenuation can be set to 0.03 first; if the model is over-fitted, it will increase; if it is under-fitted, it will decrease or set to 0.

Dropout probability: To prevent over-fitting, it can be set to 0.1 in this generation task.

Optimizer: Using the AdamW optimizer, combined with weight attenuation, it is suitable for fine-tuning. The training loss and test loss during fine-tuning are shown in Figure 3 and Figure 4. Among them, it can be seen from the test loss image that the model exhibits overfitting at step 7300, and the loss is minimized, and the effect is better at step 5900. The decision process is shown in Figure 5.

4 Design of a Motion Control System

4.1 Main hardware selection

- Mitsubishi Q series universal controller Q02U is selected as the programmable controller. Its input and output can be extended to 2048 points, 8192 points of i/o software components, 20K program capacity, and 40ns basic processing speed. The scan cycle range is 0.5 2000 Ms, and 80K program memory. USB/RS232 connection is used for communication, and the Ethernet module can also be extended.

- Qd77ms2 is selected as the simple motion module, which requires a separate slot and occupies 32 dedicated i/o channels. Linear interpolation cycle: 0.88ms. Use the SSCNet connection mode to control the servo motor, and the maximum number of controls is two axes. The servo parameters can be set on the qd77ms2 side and written into the servo amplifier or read out from the servo amplifier using sscnet. Sscnet is the bus used to connect the positioning module and the servo amplifier, also known as the servo system fieldbus network. As a common I/O-Link network, SSCNet can realize high-speed communication and simplify the system structure.

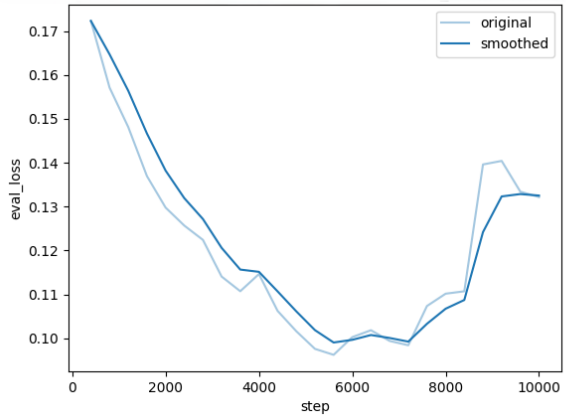

Fig. 3. training eval loss

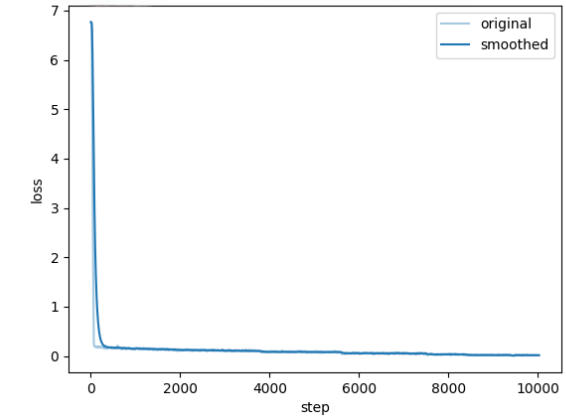

Fig. 4. training eval loss

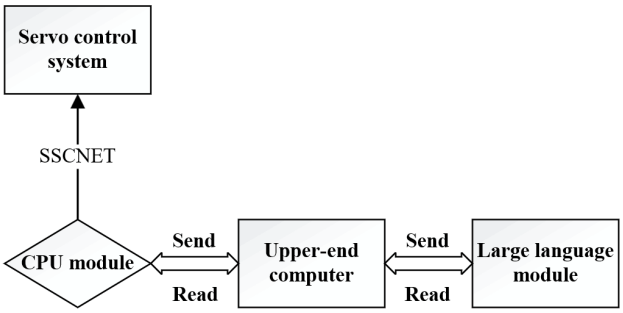

Fig. 5. Participation of a large language model in the decision-making process.

- Qj71e71-100 is selected as the Ethernet module. In this system, it is used as a communication module to connect the local PC and the upper-end computer system. Its data transmission rate can use 100Mbps (100base-tx) or 10Mbps (10baset), the communication mode is full duplex or half duplex, and TCP/IP or UDP/IP communication protocol is used.
- The servo amplifier is mr-j4-40b, The bandwidth of its current loop is 20kHz, the bandwidth of its speed loop is 2.5kHz, and the bandwidth of its position loop is 500Hz, which is connected with the servo controller through the high-speed synchronization network sscnet III/h. It can realize the real-time communication of a large amount of data between the controller and the servo amplifier. The information of the servo motor is stored in the upper information system and can be used during control. In sscnet III/h, a maximum 100m connection between stations can be carried out. Therefore, it can also correspond to large-scale systems.
- The servo motor is hg-kr43j with ultra-low inertia and small capacity; Rated speed 3000r/min, maximum speed 6000r/min, instantaneous allowable speed 6900r/min; The speed, position, and increment share a 22-bit encoder, and the resolution per revolution of the servo motor is 4194304pulses/rev.

In the servo motor positioning system controlled by PLC, the response speed is influenced by various hardware or software factors. Therefore, to enhance the response speed, we have selected the Q02U, which boasts a fast processing speed and a short scanning cycle, as the CPU. Additionally, we have chosen the QD77MS2, which features a short interpolation cycle, and the MR-J4-40B, which offers high bandwidth in the current loop, speed loop, and position loop. Furthermore, we have opted for the HG-KR43J, which possesses a small rotor inertia and a high rated speed, as the hardware component.

The equipment list obtained is shown in Table I:

Table 1: Equipment List

Equipment name	Brand	Remark
CPU module	Mitsubishi	Used
Ethernet module	Mitsubishi	Used
Serial module	Mitsubishi	reserve
Network module	Mitsubishi	reserve
Simple motion module	Mitsubishi	Used
I/o expansion module	Mitsubishi	reserve
Analog module	Mitsubishi	reserve
Servo amplifier	Mitsubishi	Used
Servo motor	Mitsubishi	Used
Low-voltage circuit	chint	Used
24V switch module	chint	Used
Switch Module	Mitsubishi	Used

4.2 I/o allocation table

Q series i/o points are allocated in hexadecimal. The starting address of the positioning module is 60H. The system uses 10 input points and 10 output points, a total of 20 points. The specific i/o point allocation is shown in Table II:

The simple motion module of Mitsubishi PLC is equipped with a buffer memory specially used for information exchange between the OKC and the motion module. The address of the module buffer memory is specified by the g value in the decimal system, as shown in Table III:

4.3 Servo system design

This system mainly realizes the operation of playing chess by receiving the data sent to the PLC by the local PC or upper computer from sscnet. Through the cooperation between the upper computer or large language model and the servo motor, and the combination of software and hardware, the Gobang game can be realized in the virtual world, providing a more intuitive and immersive game experience for players. The positioning system [11] using a simple motion module is designed as shown in Figure 6.

a) Servo motor parameter configuration: : This design adopts the method of position control and absolute positioning so that the target address can be directly input during motion positioning. The unit is set to mm. The motor resolution is 4194304puls/r, the movement is 5000um/r, the unit magnification is 1:1, and the pulse equivalent is calculated to be about 0.838mm. The conversion formula between motion distance and pulse number is:

$$P = D/0.838 \tag{3}$$

In the formula, P represents the number of pulses required for movement, and D represents the distance traveled.

Table 2: I/O allocation table

Input		Output	
Soft	**notes**	**Soft**	**notes**
X60	Qd77 ready	Y60	PLC ready
X61	Sync Flag	Y61	All-Axis Servo On
X64	Axis1M code	Y64	Axis1 stop
X65	Axis2M code	Y65	Axis2 stop
X68	Axis1	Y68	Axis1 forward jog
X69	Axis2	Y69	Axis1 reverse jog
X6C	Axis1 busy	Y6A	Axis2 forward jog
X6D	Axis2 busy	Y6B	Axis2 reverse jog
X70	Axis1 start	Y70	Axis1 locate start
X71	Axis2 start	Y71	Axis2 locate start
X74	Axis1 locate	Y74	Axis1 execution inhibit
X75	Axis1 locate	Y75	Axis2 execution inhibit

Table 3: Buffer memory address allocation table

Axis1 buffer memory		Axis2 buffer memory	
address	**notes**	**address**	**notes**
U6G800	Axis1 feed value_L	U6G900	Axis1 feed
U6G804	Axis1 feed rate_L	U6G904	Axis1 feed rate_L
U6G1500	Axis1 positioning	U6G1600	Axis1 positioning
U61502	Axis1 axis error	U6G1602	Axis1 axis error
U6G1503	Axis1 restart	U6G1603	Axis1 restart
U6G1518	Axis1 jog speed_L	U6G1618	Axis1 jog
U6G2000	Axis1 positioning	U6G8000	Axis1 positioning identifier
U6G2004	Axis1 command	U6G8004	Axis1 command speed_L
U6G2006	Axis1 positioning	U6G8006	Axis1 positioning address

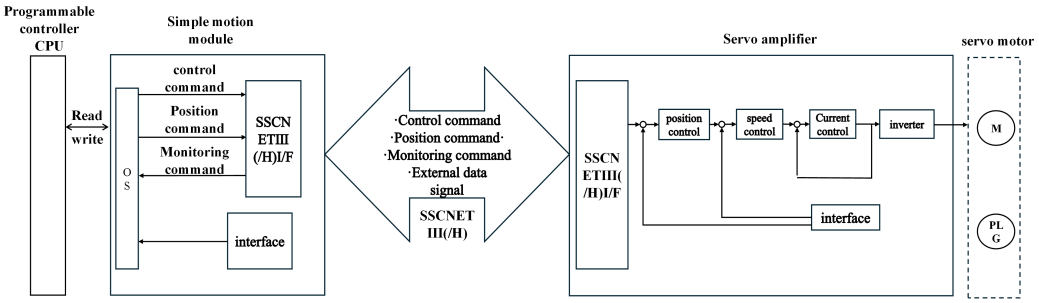

Fig. 6. Positioning System for Simple Motion Modules.

Table 4: Parameter configuration of the servo motor

Project	Axis 1 (X-axis)	Axis 2 (Y-axis)
Pr1:Unit Settings	0: mm	0: mm
Pr2:Pulses per revolution	4194304pulse	4194304pulse
Pr3:Movement per revolution	5000.0um	5000.0um
Pr4:Unit multiplier	1:1 ratio	1:1 ratio
Pr7:Bias speed at startup	0.00mm/min	0.00mm/min
Pr8:speed limit	10000.00mm/min	10000.00mm/min
Pr9:acceleration time	1000ms	1000ms
Pr10:deceleration time	1000ms	1000ms

The specific parameter settings are shown in Table IV.

b) Servo motor parameter configuration: Software travel limit: The specification of the Gobang board is 15 × 15, and each grid is 2.5cm in the longitudinal direction and 2.4cm in the transverse direction. Therefore, the size of the whole board is: 37.5cm in the longitudinal direction (but usually calculated as 35cm) and 36cm in the transverse direction. Considering the position of the chess box, the reference motion range of the manipulator can be set between -60mm 60mm.

c) Origin regression : In servo motion control, origin regression is an important step to ensure that the system is in a known starting position before each operation. An accurate reference point is the basis for ensuring subsequent motion control [12]. The origin regression mode is a data selection type, and the origin parameter settings are shown in Table V.

Table 5: Origin and Displacement Settings

P43:Origin	6:Data setting	6:Datasetting type
P44:Origin	0:Positive	0:Positive direction
P45:Origin	0.0um	0.0um
P46:Origin	300mm/min	300mm/min
P47:Cree speed	100mm/min	100mm/min
P53:Displacement	0.0um	0.0um
P55:Not return	0:Do not perform control	0:Do not perform control

d) Motion control : The board of Gobang is generally 15 × 15, and the spacing between each intersection is about 2.5 cm. In order to facilitate calculation and positioning, the point spacing of the upper computer is expressed as 25 pixels; that is, 1mm corresponds to one pixel. Every time the simulation manipulator moves a chess piece point, it needs to move 25 pixels. The movement amount of the servo motor per revolution is 5000um, so the servo motor needs to rotate 25mm/5=5 turns. Therefore, the position data of the manipulator and servo motor can be converted into the following formula:

$$J = D \times 1000 \tag{4}$$

In the formula, J is the number of pixels moved by the manipulator, and D is the position of the servo motor.

If the position of the drop is known, the first point in the upper left corner of the chessboard can be used as the reference point to calculate the distance from the origin to the reference point, and then the total displacement of the servo motor (the speed limit of the servo motor can be set to 1000.00mm/min) can be calculated according to the distance from the drop point to the reference point (how many pixels are displayed in the upper computer), as shown in Figure 7:

Take the servo X-axis as an example, if the displacement of the servo X-axis is x (mm). The horizontal distance from the reference point is a pixel. If the horizontal distance between the reference point and the falling sub point is n points, the calculation formula of the movement is:

$$x = a + n \times 25 \tag{5}$$

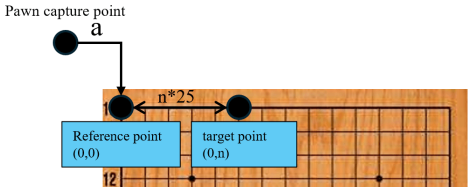

Fig. 7. Positioning System for Simple Motion Modules.

After calculating the displacement of each servo axis, the PLC can issue an absolute displacement command to the servo motor according to the preset motion control algorithm, so as to achieve accurate motion control. Since the current feed position unit in the motion module is um, the amount of direct monitoring data will be too large. In order to facilitate linkage with the manipulator in the upper computer, the data can be converted into mm, which just corresponds to the pixels of the upper computer. Every 1mm of servo motor movement, the manipulator will move one pixel relative to. It should be noted that when the system collects position data, the distance is expanded by 10 times, so when converting the unit to mm, the variable needs to be divided by 10000.

The physical hardware diagram is shown in Figure 8.

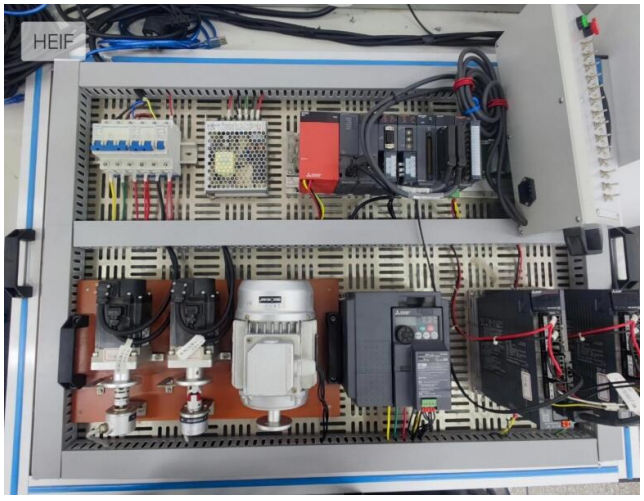

Fig. 8. Hardware object drawing

5 Conclusion

This paper presents an integrated scheme of a PLC multi-axis servo control system based on a large language model decision, and completes the design and implementation of the system accord-

ing to the scheme. This system selects Mitsubishi Q series PLC, and combines a high-performance CPU module, positioning module, Ethernet module, servo amplifier, AC servo motor, and local deployment of a large language model to build a large language decision-making system that can make accurate decisions and a multi-axis linkage servo control system that can accurately realize positioning.

Although the design and implementation of this system have made some achievements in the multi-axis servo control system, there are still some shortcomings and unknown areas to be further explored. For example, the scan cycle of the system can be further reduced. Although the current system meets the basic functional requirements, there is still room for improvement in the scanning cycle, such as reducing the scanning cycle by optimizing the algorithm and adjusting the hardware configuration.

It is hoped that the design and implementation of this system can open up new application fields for the servo system and promote the technical progress and development of the whole industry. The future of the multi-axis servo control system is full of challenges and opportunities. We look forward to solving more existing problems and exploring more unknown fields in future research.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] Zhang J, Li H, Liu Q, Li H. EtherCAT Master Design and Real-time Optimization for Multi-axis Motion Control. In: Instrument Technique and Sensor; 2024. p. 105-10.

[2] Zhou S. Multi-axis Servo Drive EtherCAT Network Control Architecture. Journal of Automation & Instrumentation. 2023:79.

[3] Zhang Y, Mu C, Lu M. Data-Based Feedback Relearning Algorithm for Robust Control of SGCMG Gimbal Servo System with Multi-source Disturbance. Transactions of Nanjing University of Aeronautics and Astronautics. 2021;38(S2):225-36.

[4] Wei M. Design and Control of a Three-Axis Motion Servo Control System Based on a CAN Bus. Energies. 2023;16(10):4208.

[5] Chen J, Tsai-Hsuan, Lee-Chia-Hua. Multi-axis Servo Control System. Journal of Industrial Electronics. 2023.

[6] Tian L. Design of a PLC-based Control System for a Small Engraving Machine; 2024. Master's thesis, 2024. DOI: 10.27776/d.cnki.gwhgy.2023.000102.

[7] Rahman AU, Alsenani Y, Zafar A, Ullah K, Rabie K, Shongwe T. Enhancing Heart Disease Prediction Using a Self-attention-based Transformer Model. Scientific Reports. 2024.

[8] Kamran M, Faizan M, Wang S, Han B, Wang WY. Generative AI and Prompt Engineering: Transforming Rockburst Prediction in Underground Construction. Buildings. 2025;15(08).

[9] Fakih M, Dharmaji R, Moghaddas Y, Araya GQ, Ogundare O, Al Faruque MA. LLM4PLC: Harnessing Large Language Models for Verifiable Programming of PLCs in Industrial Control Systems. In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering: Software Engineering in Practice; 2024. p. 192-203.

[10] Mao Y, Ge Y, Fan Y, Xu W, Mi Y, Hu Z, et al. A Survey on LoRA of Large Language Models. Front Comput Sci. 2024.

[11] Zhang Z. Research on PLC-based Control System for Three-axis Orthogonal Robot. Machinery Design & Manufacturing. 2020.

[12] Chen Y, Bi C, Yang B. Design of Servo Motor Control System. Electric Drive. 2022:68.