SIN-CHN SCIENTIFIC PRESS PTE.LTD.

Molecular & Cellular Biomechanics 2025, 22(2), 1137.
https://doi.org/10.62617/mcb1137

Article

Mechanics mechanisms and optimization strategies for the interaction between the motion precision of mechanical arms and biological tissues in medical device Ex Vivo diagnostics

Shengzhong Li[1], Yun Zhang[2,*]

[1] Guangzhou Wondfo Biotech Co. Ltd., Guangzhou 510663, China
[2] Gongdong Medical Device Qulity Surveillance and Test, Guangzhou 510663, China
*** Corresponding author:** Yun Zhang, 1953450782@qq.com

CITATION

Li S, Zhang Y. Mechanics mechanisms and optimization strategies for the interaction between the motion precision of mechanical arms and biological tissues in medical device Ex Vivo diagnostics. Molecular & Cellular Biomechanics. 2025; 22(2): 1137. https://doi.org/10.62617/mcb1137

ARTICLE INFO

Received: 17 December 2024
Accepted: 26 December 2024
Available online: 11 February 2025

Abstract: In this paper, a control system of medical robot arm based on DRL algorithm is designed by combining deep reinforcement learning (DRL) with compliant control. The system uses a trial-and-error mechanism to collect data and gradually optimize control strategies through continuous interaction between the robotic arm and the environment. Considering the actual use cost, time cost and security problems, the model is trained by the simulator based on physics engine, and the trained model is transferred to the actual robot for verification. To ensure seamless communication between different software components, Robot Operating System (ROS) was chosen as a development platform to build modular and distributed systems that are easy to test and modify. The experimental results show that the maximum distance error and repeated positioning accuracy are obviously optimized after Modified Denavit-Hartenberg (MDH) parameter modification.

Keywords: multi-degree of freedom robotic arm; in vitro diagnostic instrument; motion control accuracy; deep reinforcement learning

1. Introduction

With the continuous development of modern medicine, In Vitro Diagnostics (IVD) plays an increasingly important role in clinical treatment, disease prevention and health management [1]. IVD devices are used to collect, prepare and analyze biological samples to provide information about the health status of the human body, and are essential for the early detection of diseases, accurate diagnosis and monitoring of treatment effects. Among them, as one of the key components of the IVD instrument, the multi-degree of freedom robot arm undertakes a variety of tasks in the sample processing automation process, such as sample transfer, reagent addition, reaction mixing and so on.

To maintain the efficiency and precision of these processes, the accuracy of motion control in multi-degree of freedom robotic arms is a critical factor that influences the performance of IVD instruments. The precision of the robotic arm's movements not only impacts the speed and quality of sample processing but also directly influences the reliability of the diagnostic outcomes [2]. Therefore, optimizing the motion control accuracy of the multi- degree-of-freedom (DOF) manipulator is one of the key challenges to improve the overall performance of the IVD instrument.

In recent years, with the development of robotics, sensor technology and artificial intelligence algorithms, new ideas and technical means have been provided to solve this problem [3]. For example, by introducing high resolution encoder, force feedback system and advanced motion planning algorithm, the motion trajectory of the manipulator can be accurately controlled [4]. Utilizing machine learning and deep learning algorithms can empower the robotic arm to adapt to complex work environments and meet varying task requirements.

This study seeks to investigate optimization strategies for enhancing the motion control accuracy of multi-degree of freedom robotic arms used in medical device IVD instruments. By proposing a series of practical methods and solutions that align with current technological advancements, the aim is to boost the operational precision and service efficiency of IVD instruments. Ultimately, this research endeavors to foster scientific and technological progress in medical diagnostics, addressing the rising demand for precise medical services.

2. Correlational research

In recent years, there have been a lot of research results on motion control accuracy optimization strategies for multi-DOF manipulators. These studies can be divided into three categories according to technical means: one is based on traditional control theory; The second is the combination of machine learning and artificial intelligence algorithms. The third is the emerging hybrid control system, such as the scheme proposed in this study that fuses deep reinforcement learning (DRL) with compliance control.

Liu et al. [5] of Shenzhen Jingfeng Medical Technology Co., LTD. (Shenzhen, China), successfully developed the fifth-generation surgical robot "EDGE SP1000". Compared to traditional porous robots, the single-hole minimally invasive surgical robot only needs a small incision of 2 cm to place four serpentine operating arms into the narrow working space, and the operator can complete tasks such as peeling paper cranes, stitching egg membranes and quail eggs through the master-slave teleoperation system. Based on the technology of single-hole minimally invasive surgical robot, Xu et al. [6] of Jiaxing First Hospital successfully implemented the first single-hole robot-assisted radical resection of prostate cancer in China. This single-hole laparoscopic surgical robot independently developed by Shanghai Jiao Tong University marks another important progress in the field of surgical robots in China. In order to improve the stability in teleoperation, Lin et al. [7] introduced a wavelet-based generalized learning adaptive filter designed to predict and mitigate physiological tremors. This method can extract multi-dimensional coupling information from the wavelet domain by redesigning the width learning system structure. Meanwhile, Liu et al. [8] developed a coupled model combining fuzzy logic, wavelet transform and artificial neural network to analyze hand tremor in microsurgical robotic systems, especially the response to burst signals, taking into account both time and frequency domain features. Alebooyeh and Urbanic [9] presented a technique that employs multi-layer perceptron neural networks for predicting the inverse kinematics of 7-DOF robotic systems, achieving a prediction confidence level within the range of 90% to 95%. To enhance the robustness and accuracy of robotic arm control, Ying Liu et al. [10] developed a self-tuning particle swarm optimization (PSO) fuzzy PID positioning controller, building upon traditional fuzzy PID control mechanisms. By refining the quantization and scale factors within the PSO algorithm, they were able to improve the overall robustness and precision of the robotic arm's movements. Julian Nubert et al. [11] introduced an innovative robust model predictive control (MPC) method that ensures reliable and safe tracking of dynamic setpoints while maintaining stability and satisfying operational constraints. The effectiveness of this approach was substantiated through experimental validation. Maria Vittoria et al. [12] addressed challenges related to balancing position and orientation tasks in MPC planning by devising a method to efficiently manage the contact forces exerted by the end-effector. This solution was tested through hardware experiments on practical tasks such as opening doors, demonstrating the efficacy of their controller design. Further, Chelsea Finn et al. [13] designed a deep space autoencoder that can learn state representations directly from camera images to construct state Spaces, enabling the robot to dynamically manipulate objects in the environment through closed-loop control. Rong Jiang et al. [14] proposed a vision-based deep reinforcement learning method that accelerates the learning process and improves performance by using asymmetric inputs and adding state prediction auxiliary tasks to the reinforcement learning model, thereby optimizing the motion control of a robotic arm. Ryosuke Tsumura et al. [15] proposed an automated robotic arm platform for remote auscutation, which can obtain external body information through a lidar camera, autonomically locate the stethoscope, and ensure safe and flexible contact while controlling the contact force within a certain range. This enables the platform to perform relatively simple and safe diagnostic tasks.

These studies significantly advance the field of medical equipment robot technology, offering robust technical support for the future development of personalized and precision medicine. By improving the accuracy, adaptability, and reliability of robotic systems, they pave the way for more tailored and effective medical treatments.

3. Method

3.1. Deep reinforcement learning theory and compliant control theory

3.1.1. Basic principle

Deep Reinforcement Learning (DRL)

Deep reinforcement learning is a domain within machine learning that integrates the representation abilities of deep learning with the decision-making features of reinforcement learning. In DRL, agents learn strategies by interacting with the environment to maximize cumulative rewards. The core idea is that through the process of trial and error, the agent gradually optimizes its behavior pattern to reach the optimal solution.

Markov decision process (MDP): DRL is usually modeled based on Markov decision process, where state $s \in S$, action $a \in A$, transition probability $P(s'|s,a)$, and immediate reward $R(s,a)$ form a complete system description.

Bellman equation: To evaluate how good or bad a strategy is, use the value function $V^{\pi}(s)$ or the action value function $Q^{\pi}(s,a)$, which satisfy the Bellman expectation equation:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[R(s,\pi(s)) + \gamma V^{\pi}(S')|S = s] \tag{1}$$

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[R(s,a) + \gamma Q^{\pi}(S',A')|S = s, A = a] \tag{2}$$

Where γ is the discount factor, $0 < \gamma \leq 1$, which balances the importance of current and future rewards.

Compliant Control

Compliant control is designed to allow robots to safely interact with objects in uncertain or dynamic environments. By adjusting the rigidity or compliance of the end effector, a more natural and safe human-machine collaboration can be achieved.

Impedance Control: This method adjusts the behavior of the end effector by defining a virtual spring-mass-damping system so that the robot responds to external forces similar to the elastic deformation of a physical system. The basic formula is:

$$F = m\ddot{x} + b\dot{x} + kx \tag{3}$$

In this context, F represents the externally applied force, x is the displacement, m denotes the mass, b is the damping coefficient, and k stands for the spring constant.

Admittance Control: This method in turn defines the force-velocity relationship, i.e. given a force input, the corresponding velocity output is calculated. Its expression is as follows:

$$v = \frac{1}{m}F - \frac{b}{m}\dot{x} - \frac{k}{m}x \tag{4}$$

3.1.2. The combination of deep reinforcement learning and compliant control

Combining deep reinforcement learning with compliant control can create a more intelligent and flexible control system, which is especially suitable for applications that require high adaptability and robustness, such as medical surgery, rehabilitation training and other fields.

Learning compliance controller: The DRL framework is used to design a controller that can self-learn the optimal compliance characteristics. For example, the parameters m, b, and k in impedance control can be dynamically adjusted through reinforcement learning to give the best performance in different environments.

Fusion Solution Based on Model Predictive Control (MPC): this solution integrates the strengths of MPC and DRL. It leverages the short-term planning capabilities of MPC to guarantee system stability and safety. Simultaneously, it utilizes the learning mechanisms of DRL to continuously optimize the overall performance of the system over the long term. This combination aims to harness the precision and predictability of MPC with the adaptive and strategic optimization abilities of DRL, providing a robust approach to complex control problems.

End-to-end learning architecture: Build an end-to-end mapping from sensor data to control instructions, and through a large number of simulations and real-world experiments, let agents learn how to adapt their compliance characteristics to changes in the environment.

The combination of deep reinforcement learning and compliant control theory not only expands the application boundary of traditional control theory, but also provides a new possibility for developing a new generation of intelligent and adaptive robot systems.

3.2. Kinematics analysis of robotic arm

3.2.1. D-H parameter modeling

The operation of the manipulator relies on a specific mechanism to realize the movement of parts and tools in space, which inevitably involves the expression of the position and attitude of parts, tools and mechanisms. To accurately depict the pose changes of the robotic arm, a system that includes a world coordinate system will be utilized. In establishing the manipulator's coordinate system, both the base coordinate system and the joint coordinate systems are essential components. The base coordinate system serves as the foundation for defining all other coordinate systems and clearly describes the relative position and orientation between the joints and the end-effector of the robotic arm. This setup ensures precise tracking and control of the manipulator's movements within its operational environment.

The robotic arm utilized in this study is the UR5e, a six-degree-of-freedom serial-type manipulator featuring six rotary joints. For the setup of its coordinate system, the origin of the base coordinate system (equivalent to the Cartesian coordinate system) is first established on the robot arm's base. The X, Y, and Z axes are then defined according to the front, side, and vertical orientations relative to the robot arm. Following the establishment of the base coordinate system, each joint's center is designated as the origin for its respective coordinate system following the D-H modeling principle. This process involves defining the coordinate system for each joint of the robot arm sequentially. Each joint's coordinate system is illustrated in **Figure 1**, where parameters such as the link twist angle α_i, link length l_i, joint angle θ_i, and offset d_i are specified, along with the motion range of each joint. These parameters are summarized in the D-H parameter table, as shown in **Table 1**.

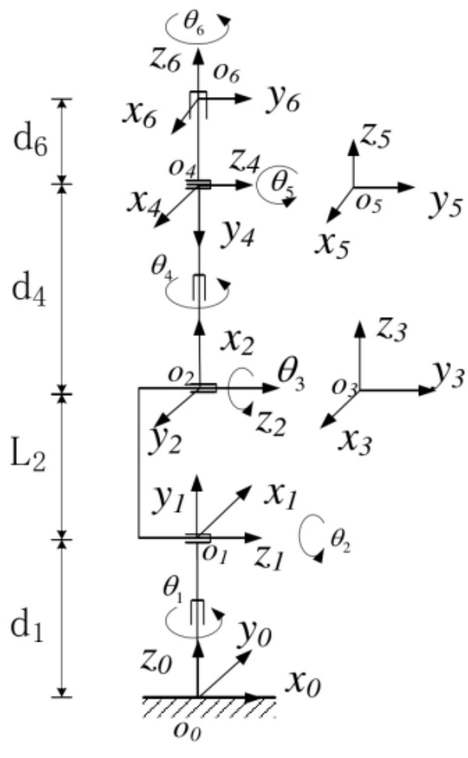

Figure 1. The D-H coordinate system of the manipulator.

Table 1. Mechanical arm D-H parameter table.

i	θ_i	d_i	l_i	α_i	Range of motion
1	90°	258	0	90°	10°~340°
2	90°	0	400	0°	−60°~240°
3	90°	0	0	90°	−60°~240°
4	0°	360	0	-90°	−160°~160°
5	0°	0	0	90°	−140°~140°
6	0°	295	0	0	−160°~160°

After the determination of all linkage parameters and the establishment of coordinate system, the base coordinate system is redefined as $\{0\}$, and the coordinate system of the *I*-th joint is marked as $\{i\}$. Using the homogeneous transformation matrix ${}^{i-1}_{i}T$ to realize the transformation between the coordinate system $\{i\}$ and the coordinate system $\{i–1\}$, the following transformation matrix can be obtained:

$$\begin{aligned} {}^{i-1}_{i}T &= R_Z(\theta_i)D_Z(d_i)D_X(a_i)R_X(\alpha_i) \\ &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & s\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -ds\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5}$$

3.2.2. Forward kinematics of the manipulator

Forward kinematics analysis for a robotic manipulator examines the relationship between the position and orientation (pose) of the end-effector and the joint angles. Specifically, it involves solving for the pose of the end-effector given known joint angles. This process is fundamental to the motion planning and control of the robot arm, as it allows for precise control by determining the exact location and orientation of the robot arm in three-dimensional space. The key steps involved in forward kinematics analysis are as follows:

1) Establishing the Kinematic Model: The first step is to define the kinematic model of the manipulator. This includes specifying the D-H parameters that describe the axis of each joint and the geometric relationships between successive joints.
2) Determining Joint Variables: Next, the angle or displacement for each joint is measured or calculated. These measurements serve as the joint variables that will be used in subsequent calculations.
3) Calculating Transformation Matrices: Using the D-H parameters and the determined joint variables, the homogeneous transformation matrix for each joint is computed. Each matrix represents the spatial transformation from one joint to the next, encapsulating both rotation and translation.
4) Obtaining the Total Transformation Matrix: By multiplying the individual transformation matrices of all joints, the total transformation matrix from the base coordinate system to the end-effector coordinate system is derived. This matrix provides the complete transformation information necessary to determine the end-effector's specific position and orientation relative to the base.
5) Verification of Results: Finally, the accuracy of the forward kinematics solution is verified through simulation or real-world testing. This step ensures that the

theoretical calculations align with the actual performance of the robot arm, confirming the reliability of the kinematic model and the precision of the motion control.

This methodical approach to forward kinematics not only facilitates accurate and reliable robotic operation but also serves as a cornerstone for advanced robotic applications requiring high precision and adaptability.

3.3. Solution of inverse kinematics of robotic arm

Unlike forward kinematics, inverse kinematics in a robotic arm focuses on determining the angles or displacements of each joint based on the specified end-effector pose. The methods of inverse kinematics analysis mainly include analytical method, iterative method and geometric method. Typically, inverse kinematics analysis involves the following steps:

1) Build a kinematic model: As with forward kinematic analysis, this step needs to determine the D-H parameters that describe the structure of the manipulator.
2) Determine the Pose of the End Effector: The target position and orientation (pose) information of the end effector can be established through manual setting or alternative methods.
3) Solving for Joint Variables: Utilizing the 16 elements in the transformation matrix ${}_6^0T$ from the base coordinate system to the end effector, the precise angles or displacements of each joint are calculated using inverse kinematics algorithms. This transformation matrix encapsulates all the positional and orientational data of the end effector relative to the base, enabling the determination of the necessary joint configurations to achieve the desired end-effector pose.

$$\begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_1^0T\,{}_2^1T\,{}_3^2T\,{}_4^3T\,{}_5^4T\,{}_6^5T \tag{6}$$

In the process of solving joint variables, the inverse transformation of the position matrix is usually left multiplied by the corresponding matrix equation to separate and finally solve each joint variable. However, in some cases, the inverse kinematics problems show nonlinear characteristics, which makes it impossible to obtain analytical solutions directly. In this case, it is necessary to adopt numerical methods, such as iterative method or numerical optimization techniques, to gradually approximate and solve the joint variables. This method can deal with the complex structure of the manipulator, and can effectively calculate the joint configuration that satisfies the specified pose of the end-effector even in the absence of a clear analytical solution. The flexibility and adaptability of numerical methods make them an important tool for solving nonlinear inverse kinematics problems.

4) Verification results: It is possible to verify whether the calculated joint Angle can achieve the expected end-effector position and attitude through simulation or actual operation of the robot arm.

Before solving the inverse kinematics, it is first necessary to confirm whether the obtained position coordinates of the end of the manipulator are located in its working space, so as to ensure the feasibility and stability of the solution. Only when the target position is within the reachable range will the inverse solution continue to be calculated. Since the inverse kinematics problem may have multiple solutions, some of which may be physically infeasible or unstable, various constraints must be considered in the solution process to screen out the optimal solution that meets both the physical constraints of the robot arm and the task requirements.

4. Application of model

4.1. Design and construction of manipulator control system

The core of this scheme is to design a medical robot control system based on DRL algorithm, which aims to make the robot arm can efficiently complete various medical tasks. The system utilizes deep reinforcement learning to collect data and gradually optimize control strategies through a trial-and-error mechanism through continuous interaction between the robotic arm and the environment. However, considering the actual use cost of the robot arm, the cost of time, and the safety of interacting with the environment, it is not realistic to conduct a large number of tests directly in the real environment. Therefore, with the help of physics engine-based emulators such as PyBullet or MuJoCo, an efficient alternative is provided. Scholars can train the model in these simulation experiment environments, and then transfer the trained model to the actual robot for verification.

To ensure seamless communication between different software components and to build a modular and distributed System that is easy to test and modify, we chose ROS as our development platform. ROS is an open source robotic operating system that not only provides standardized messaging mechanisms, supports interoperability across programming languages and operating systems, but also simplifies the integration and management of complex systems. Therefore, this article will deploy and test the trained model on Linux systems and ROS environments to evaluate its performance and applicability. The overall architecture is shown in **Figure 2**, showing the complete process from simulation training to practical application.

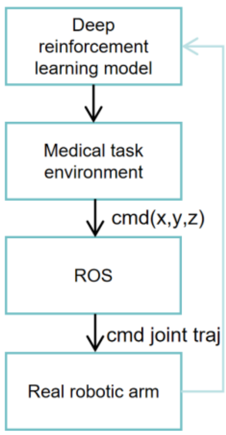

Figure 2. Overall architecture drawing.

4.2. Hardware and software integration

The manipulator control system consists of two parts, hardware and software, which work together to ensure that the manipulator can achieve high-precision and efficient movement. The control system hardware of UR5e robot arm mainly includes controller, motor, sensor, tool, communication module, power supply and signal line and other components. Among them, the controller, as the core component, is responsible for receiving data from the sensor and issuing instructions to the actuator. UR5e is based on Linux system UR controller, built-in special robot arm control software, can directly connect and control the UR5e robot arm, support intuitive programming and operation. The motor is used to drive the various joints of the robot arm, and six AC servo motors precisely control the speed and position to ensure the accuracy of the movement of the robot arm. Sensors such as vision, torque and tactile sensors are used to sense the surrounding environment and convert the information into electrical signals for transmission to the controller, adapting to more complex application scenarios. The choice of tools is crucial to the completion of the task, UR5e can be equipped with a variety of tools such as claws, electric wrenches, suction cups, to meet the needs of different tasks. The communication module enables the controller to exchange data with other external devices. Common modules include Ethernet interfaces, etc., to ensure the interconnection of the system. Power and signal lines provide the necessary power and support for the entire system, and ensure the reliability and stability of the system through carefully arranged lines.

In terms of software, the control system of the UR5e robot arm includes three main parts: motion control algorithm, motion planning algorithm and user interface, ensuring that the operation is accurate and convenient. UR programming software provides a visual programming environment, users can easily create complex motion programs by dragging and clicking, while supporting scripting languages such as Python, convenient custom development for advanced users. ROS as a powerful robot application platform, it not only simplifies programming and control, but also provides a rich function library and tools, which greatly promotes the development efficiency. Software Development Kit (SDK) covers support for several programming languages, such as C, Python, and Java, so users can choose the right development tool for their skills and project needs. The SDK contains rich apis and sample code to help developers get started quickly. The Web console provides a way to access the robot arm through a browser, and users can monitor and adjust the status of the robot arm anywhere in real time, set parameters, and even debug it, greatly improving the flexibility of use.

4.3 Experimental results and analysis

Manipulator calibration refers to the use of higher precision measurement tools to evaluate and adjust the position and attitude information of the manipulator, and then establish an accurate calibration model. This process employs parameter identification technology to precisely determine the structural parameters of the robotic arm model, thereby enhancing its absolute positioning accuracy. To improve the absolute motion accuracy of the manipulator, the research adopts a method of calibrating and identifying kinematic parameters. Specifically, a Leica-AT901 laser tracker is utilized as the measurement tool. This device leverages laser tracking technology to accurately acquire the position and orientation data of the robotic arm's end effector, ensuring precise and reliable measurements that contribute to refining the manipulator's performance.

4.3.1. Kinematic calibration

Kinematic calibration of a manipulator arm based on the MDH method involves the use of improved Denavit-Hartenberg parameters to accurately describe and adjust the geometry of the manipulator. By introducing a more flexible definition of coordinate system, this method can more accurately represent the relative position and attitude between the joints, thus improving the accuracy of the kinematic model. In the calibration process, the first step involves acquiring the actual pose data of the manipulator's end-effector using a high-precision measurement tool, such as a laser tracker or an optical measurement system. These tools provide accurate position and orientation data essential for calibrating the robotic arm's kinematic parameters and improving its overall accuracy. Then, by comparing the difference between the actual measured values and the predicted values of the theoretical model, an optimization algorithm is used to identify and correct the errors in the MDH parameters. This process not only corrects the geometric parameters of the manipulator, but also compensates for possible non-geometric errors, such as joint clearance and deformation. Ultimately, calibrated MDH parameters can significantly improve the absolute positioning accuracy of the robotic arm, ensuring its reliability and accuracy in complex tasks.

In the MDH parameter identification experiment, a total of 375 sets of data were collected, each set of data recorded the corresponding joint Angle, and the error between the measured value and the theoretical value of the end-effector position was calculated. In the identification process, the first 365 sets of data were used for parameter identification calculation to determine the variation of MDH parameters. Then, the remaining 10 groups of data were used to verify the identification results, as shown in **Tables 2** and **3**.

Table 2. Identify previous MDH parameter values.

i	θ_i	d_i	l_i	α_i	β
1	90°	260.743	0	90°	0°
2	90°	1.839	399.631	0°	
3	90°	0	0	90°	
4	0°	352.768	0	−90°	
5	0°	0	0	90°	
6	0°	211.537	0	0	

Table 3. The identified MDH parameter value.

i	θ_i	d_i	l_i	α_i	β
1	89.9542°	263.5376	0.1456	90.0043°	0.0164°
2	89.7843°	1.6321	400.1758	0.0164°	
3	89.9639°	0.1098	0.7424	89.9521°	
4	0.3468°	351.5762	−0.1776	−90.4869°	
5	0.7165°	0.9771	0.0001	90.6432°	
6	0°	211.5476	0.0001	0	

In order to verify the identification results, the last 10 data points in the above measurements were selected for verification. The MDH parameters before and after identification were used to calculate the theoretical values of each position, and then the actual position errors of these sampling points before and after identification were compared, as shown in **Table 4**. In this way, the improvement effect of the identification process on the position accuracy can be quantified.

Table 4. Comparison and verification of sampling point distance error before and after DH correction.

Validation parameter	Before correction	After revision
Maximum error (mm)	6.1928	2.7613
Root mean square error (mm)	3.8192	1.1237

Table 4 presents the maximum distance error and root mean square error before and after the correction of sampling points. The data indicates that, following the modification of the MDH parameters, there is a significant improvement in the absolute positioning accuracy of the robot. Specifically, the maximum distance error decreased from 3.7970 mm to 1.4108 mm. This reduction demonstrates that optimizing the MDH parameters markedly enhances the robot's positioning accuracy.

This improvement is not only reflected in the reduction of the maximum error, but also reflected in the improvement of the overall accuracy level, which proves the effectiveness and importance of parameter correction.

4.3.2. Repeated positioning accuracy measurement

Repeated positioning accuracy is a critical metric for assessing robot performance. In this study, Group 7 of the validation sampling points was chosen to conduct repeated positioning accuracy tests. The procedure involved having the robotic arm's joint reach a specified position with a fixed attitude 20 times under identical conditions. Each time, the actual position of the robot's end effector was measured. The error distribution for each subsequent movement was then calculated relative to the first measurement result, as illustrated in **Figure 3**.

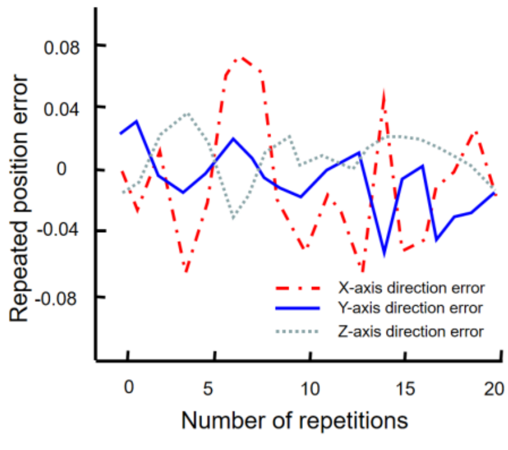

Figure 3. Repeated positioning accuracy error of mechanical arm.

The organized results are presented in **Table 5**. The data indicate that the mechanical arm's repeated positioning accuracy is as follows: less than 0.076 mm in the *X* direction, less than 0.053 mm in the *Y* direction, and less than 0.04 mm in the *Z* direction. Overall, the repetitive positioning accuracy is maintained within 0.0742 mm, fully satisfying the stringent requirements for the mechanical arm's repeated positioning precision.

Table 5. Repeat the positioning accuracy measurement results.

Measurement parameter	***X*-axis direction**	***Y*-axis direction**	***Z*-axis direction**	**Combined error**
Maximum repeatability error (mm)	0.076	0.053	0.04	0.0742
Root mean square error (mm)	0.0438	0.0216	0.0159	0.0425

In the process of measurement, the laser tracker itself may have some measurement errors, and the robot rotation joint also has reverse clearance errors. These factors together affect the repetitive positioning accuracy of the robot arm and constitute the total repetitive position error. Nevertheless, the experimental results still show that the robotic arm performs well in repeated positioning in practical applications and can stably meet the expected accuracy standards.

The optimization strategy proposed in this study, especially the control system design based on DRL, not only improves the motion control accuracy of the multi-DOF manipulator, but also significantly improves the work efficiency and service quality of the IVD equipment. For example, in large-scale medical testing scenarios, the improved robot arm can shorten the detection time and reduce the cost by processing samples more quickly and accurately. In addition, high-precision manipulator motion control helps to reduce the misdiagnosis rate, which in turn improves the treatment effect of patients. In the long run, such technological innovations will also promote the efficient allocation of medical resources and bring more well-being to society.

In addition to its application in the field of in vitro diagnostics, the method of combining DRL with compliance control proposed in this study is also applicable to molecular and cellular biomechanics. For example, extremely high spatial resolution and operational precision are required in cellular micromanipulation, such as single cell capture and intracellular material transfer. The control system developed in this study is able to provide the necessary compliance and stability to ensure that the cellular structure is not damaged during operation. In addition, in tissue engineering, the construction of biological scaffolds also requires precise control of the location and orientation of material deposition. With the help of the learning ability of the DRL algorithm, the manipulator can automatically adjust its parameters to meet the requirements of different tasks in the changing operating environment. Therefore, this study not only brings new solutions for IVD devices, but also provides strong technical support for cell biology and tissue engineering.

5. Conclusions

In this study, by combining deep reinforcement learning (DRL) and compliant control theory, the motion control accuracy of multi-degree-of-freedom robotic arm in medical device in vitro diagnostic instrument was successfully improved. We utilized high-precision measurement tools to acquire the position and orientation data of the manipulator arm's end-effector and employed optimization algorithms to correct errors in the MDH parameters. The experimental outcomes demonstrate a significant enhancement in both the absolute positioning accuracy and the repeated positioning accuracy of the manipulator. These findings not only validate the effectiveness of our proposed approach but also highlight its reliability and stability in real-world applications. This advancement provides robust technical support for the future development of personalized and precision medicine, ensuring more accurate and dependable robotic assistance in medical procedures.

6. Future research perspectives

In order to further promote the direction of this research, we suggest the following in-depth research:

1) Optimizing Deep Reinforcement Learning algorithms: We will continue to optimize DRL algorithms to improve their learning efficiency and generalization ability for the uncertainties existing in complex medical environments.
2) Multi-robot Collaboration: Explore ways to efficiently collaborate between multiple robotic arms, especially in tasks involving fine manipulation such as cell transplantation or multi-point sampling.
3) Interdisciplinary collaboration: Strengthen collaboration with other disciplines, such as biomedical engineering, computer science, etc., to jointly solve key problems in the application of robotic arms in the medical field.
4) Clinical validation: In future work, more attention should be paid to data collection and analysis in clinical trials to verify the actual effect of the proposed method and to continuously adjust and improve the system performance based on feedback.

Author contributions: Conceptualization, SL and YZ; methodology, SL; software, SL; validation, SL, YZ; formal analysis, SL; investigation, SL; resources, SL; data curation, SL; writing—original draft preparation, SL; writing—review and editing, SL; visualization, SL; supervision, SL; project administration, SL; funding acquisition, YZ. All authors have read and agreed to the published version of the manuscript.

Ethical approval: Not applicable.

Conflict of interest: The authors declare no conflict of interest.

References

1. Li X, Xiao J, Zhao W, et al. Multiple peg-in-hole compliant assembly based on a learning-accelerated deep deterministic policy gradient strategy[J]. Industrial Robot: the international journal of robotics research and application, 2022, 49(1): 54-64.
2. Zhao, Yufei, Shixiao Xu, and Hua Duan. "HGNN− BRFE: Heterogeneous Graph Neural Network Model Based on Region Feature Extraction." Electronics 13.22 (2024): 4447.
3. Zhao Y, Wang S, Duan H. LSPI: Heterogeneous Graph Neural Network Classification Aggregation Algorithm Based on Size Neighbor Path Identification[J]. arXiv preprint arXiv:2405.18933, 2024.

4. Y. Zhao, H. Liu and H. Duan, "HGNN-GAMS: Heterogeneous Graph Neural Networks for Graph Attribute Mining and Semantic Fusion," in IEEE Access, doi: 10.1109/ACCESS.2024.3518777.
5. Liu C, Lai C, Yao X, et al. Robot-assisted nephrectomy using the newly developed EDGE SP1000 single-port robotic surgical system: a feasibility study in porcine model[J]. Journal of Endourology, 2020, 34(11): 1149-1154.
6. Xu K, Zhao J, Fu M. Development of the SJTU unfoldable robotic system (SURS) for single port laparoscopy[J]. IEEE/ASME Transactions on Mechatronics, 2014, 20(5): 2133- 2145.
7. Lin J, Liu Z, Chen C L P, et al. A wavelet broad learning adaptive filter for forecasting and cancelling the physiological tremor in teleoperation[J]. Neurocomputing, 2019, 356: 170- 183.
8. Liu Z, Mao C, Luo J, et al. A three-domain fuzzy wavelet network filter using fuzzy PSO for robotic assisted minimally invasive surgery[J]. Knowledge-Based Systems, 2014, 66: 13-27.
9. Alebooyeh M, Urbanic R J. Neural network model for identifying workspace, forward and inverse kinematics of the 7-DOF YuMi 14000 ABB collaborative robot[J]. IFAC PapersOnLine, 2019, 52(10): 176-181.
10. Liu Y, Jiang D, Yun J, et al. Self-tuning control of manipulator positioning based on fuzzy PID and PSO algorithm[J]. Frontiers in Bioengineering and Biotechnology, 2022, 9: 1443.
11. Nubert J, Köhler J, Berenz V, et al. Safe and fast tracking on a robot manipulator: Robust mpc and neural network control[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 3050-3057.
12. Minniti M V, Farshidian F, Grandia R, et al. Whole-body mpc for a dynamically stable mobile manipulator[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3687-3694.
13. Finn C, Tan X Y, Duan Y, et al. Deep spatial autoencoders for visuomotor learning[C]//2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2016: 512-519.
14. Jiang R, Wang Z, He B, et al. Vision-Based Deep Reinforcement Learning For UR5 Robot Motion Control[C]//2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). IEEE, 2021: 246-250.
15. Tsumura R, Koseki Y, Nitta N, et al. Towards fully automated robotic platform for remote auscultation[J]. The International Journal of Medical Robotics and Computer Assisted Surgery, 2023, 19(1): e2461.