Research on trajectory optimization and adaptive control of manipulator based on deep reinforcement learning

BoLi Zhang
Anhui Agricultural University, Hefei, China
1727773672@qq.com

Abstract. In this paper, the trajectory optimization and adaptive control method of manipulator based on deep reinforcement learning (DRL) is studied, aiming at solving the contradiction between real-time, safety and adaptability of manipulator in complex dynamic environment. Aiming at the limitations of traditional control methods under unstructured disturbances, a hierarchical control architecture is proposed, which includes a high-level trajectory optimization layer and a low-level adaptive control layer, and integrates the safety RL mechanism. High-level uses time-series convolutional network (TCN) combined with attention mechanism to process time-series environmental information and generate trajectory parameters that meet security constraints; The parameter adaptive module of sliding mode control is designed at the lower level, and the stability is guaranteed by Lyapunov function. Experiments were carried out on a 7-degree-of-freedom Franka Emika manipulator platform. The results show that the proposed method has the shortest trajectory length, the lowest average tracking error and the lowest collision rate (2.1%) in dynamic scenes, the fastest planning speed and the lowest energy consumption, and is significantly superior to traditional model predictive control (MPC), DRL without safety constraints and fixed sliding mode control methods. The ablation experiment further verified the contribution of each module to the system performance, and the integrated performance of the complete system was the best, achieving the lowest tracking error, collision rate and reasonable planning time.

Keywords: deep reinforcement learning; trajectory optimization; adaptive control; manipulator; time-series convolutional network; attention mechanism.

1. Introduction

The application of manipulator is expanding from the traditional structured environment to a more complex dynamic environment, especially in the aspects of man-machine integration and obstacle avoidance. However, the traditional control method of manipulator faces the problems of response lag and energy consumption fluctuation in the face of unstructured disturbance, and the existing technical routes-model-driven and data-driven methods also have obvious limitations. Model-driven methods such as A algorithm, RRT sampling and model predictive control (MPC) are suitable for static scenes, but they face the problems of modeling error, high computational complexity and lack of environmental adaptability. Data-driven methods, especially deep reinforcement learning (DRL), also have shortcomings in sample efficiency, security and generalization ability. Therefore, the development of intelligent control framework with real-time decision-making ability and environmental adaptability has become a key problem to be solved urgently in the field of robotics.

DRL combines the feature extraction ability of deep learning with the decision optimization ability of RL, which makes it perform well in the control task of manipulator in complex environment [1]. DQN, Policy Gradient and other algorithms have been successfully applied to solve the motion planning and control problems of manipulator [2]. These algorithms learn the optimal strategy by interacting with the environment, which improves the working efficiency and adaptability of the manipulator [3]. Trajectory optimization aims to generate a path that meets specific performance indicators, such as shortest time and minimum energy consumption [4]. Traditional methods such as analytical method and numerical optimization method may be difficult to deal with complex constraints in some cases. The trajectory optimization method based on DRL, such as using the improved golf optimization algorithm, effectively avoids falling into the local optimal solution by introducing chaotic mapping and adaptive weight factors, and improves the global optimization ability of trajectory planning [5-6]. Adaptive control technology enables the manipulator to adapt to the changes of environment and task requirements [7]. Adaptive control methods based on DRL, such as adaptive sliding mode robust control, not only improve the control accuracy and robustness of the manipulator, but also enhance its learning ability, so that it can adjust the control strategy in real time to adapt to uncertain factors during the control process [8-9]. The coordinated control of multi-manipulator system is a research hotspot in the field of robotics. The research involves robot base coordinate system calibration, coordinated system dynamics modeling, controller synthesis method and other issues [10]. The research progress of coordinated control of multi-manipulator mainly focuses on the establishment of dynamic model of the system and the control synthesis of coordinated motion [11].

Aiming at the contradiction between real-time, safety and adaptability in manipulator control, this paper proposes a hierarchical control architecture based on DRL. The architecture includes high-level trajectory optimization layer and low-level adaptive control layer, and integrates security RL mechanism. High-level adopts time series convolutional network (TCN) combined with attention mechanism to shorten the time of dynamic obstacle avoidance planning; The parameter adaptive module of sliding mode control is designed at the low level, and the stability is guaranteed by Lyapunov function. The safety RL framework sets collision avoidance as a hard constraint, which reduces the training collision rate.

2. Overall architecture design

A hierarchical RL framework, as shown in Figure 1, is adopted, which is divided into a high-level DRL trajectory optimization layer and a low-level adaptive control layer. The high-level DRL trajectory optimization layer processes the time sequence environment information based on TCN-attention mechanism and generates trajectory parameters that meet the security constraints, which are guaranteed by the constraint RL (CRL) hard constraint mechanism in the security layer. In the lower layer, Lyapunov stability is realized by sliding mode adaptive controller, and disturbance compensation is realized by RL online parameter adjustment. The whole architecture ensures the security and stability of the system on the basis of real-time environmental awareness [12-13].

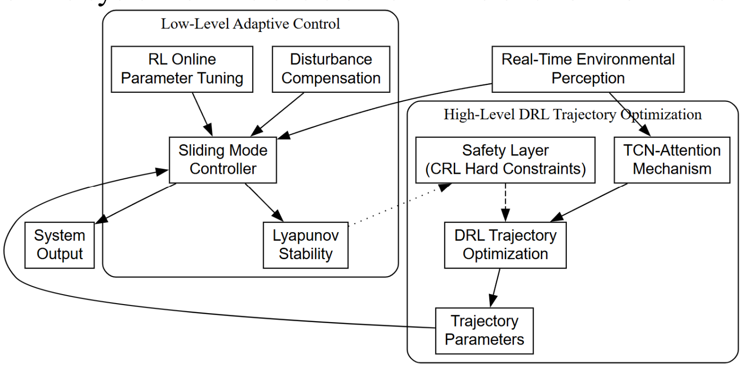

Figure 1 Hierarchical RL framework

3. Realization of high-level trajectory optimization layer

3.1 State space design

The state space of the high-level trajectory optimization layer is defined as:

$$s_t = [q_t, \dot{q}_t, O_t, g] \tag{1}$$

Among them, $q_t \in R^n$ represents the angle of each joint at the current moment and reflects the configuration state of the manipulator (n is the number of degrees of freedom); $\dot{q}_t$ stands for joint angular velocity, which is used to describe the dynamic characteristics of motion; O_t is the environmental information of obstacles, and the distribution of obstacles in the surrounding three-dimensional space is captured through the point cloud characteristics obtained by VoxelGrid coding [14]; $g \in R^3$ is the expected pose (position coordinate) of the target end effector, which is generated as the task target guidance trajectory.

3.2 Action space design

Action space is defined as:

$$a_t = [\Delta \xi_{1:T}] \tag{2}$$

That is, the offset sequence that corrects the trajectory waypoint in Cartesian space, and the time range is the next T steps. Each $\Delta \xi_k$ corresponds to the adjustment of a trajectory point, and finally an optimized safety trajectory is output. Choosing the time domain length of $T = 5$ ensures the real-time response ability, taking into account the foresight and safety of trajectory planning, so that the system can make more reasonable obstacle avoidance and path selection in complex environment 15-16].

3.3 Network structure

The high-level network adopts the architecture based on the combination of TCN and attention mechanism, and the specific process is shown in Figure 2 below.

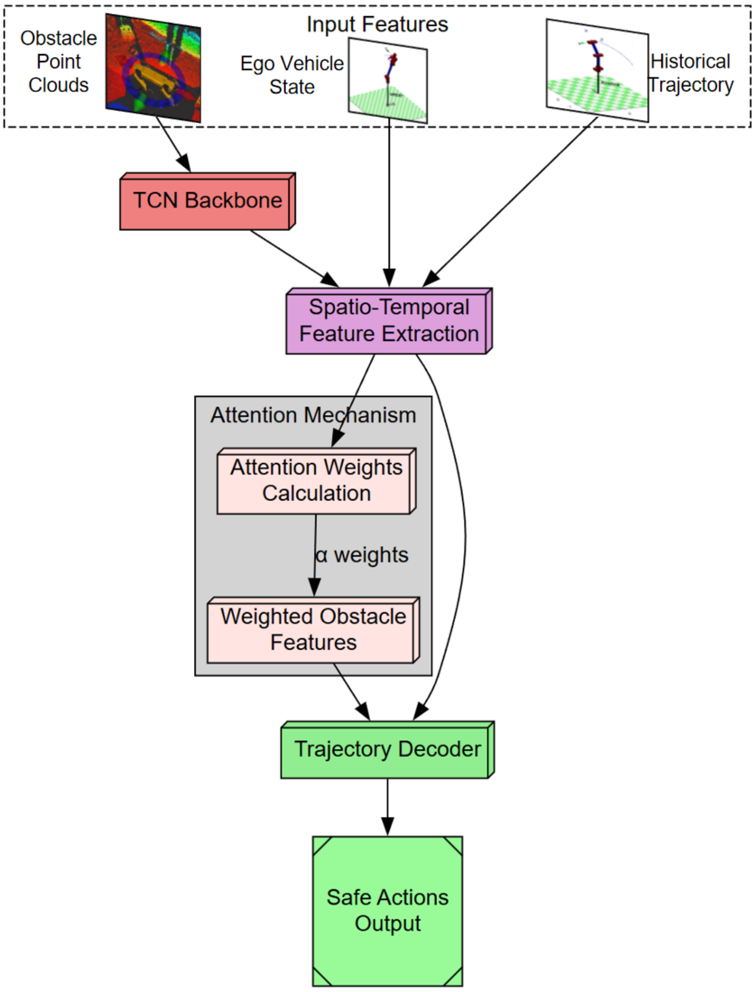

Figure 2 Architecture based on the combination of TCN and attention mechanism

TCN backbone network is used to extract the spatio-temporal characteristics of state input, especially to process the obstacle point cloud data O_t in time series and capture its dynamic changing trend [17]. The spatio-temporal feature extraction module further enhances the interactive modeling ability between multimodal state information; The attention module calculates the weight α_i of each obstacle feature h_i, and the formula is:

$$\alpha_i = \frac{\exp(W_a h_i)}{\sum_j \exp(W_a h_j)} \tag{3}$$

W_a is a learnable parameter, which is used to measure the influence of different obstacles on the current trajectory, so as to realize the priority attention to key obstacles; Obstacle weight α weights attention results to the original trajectory and guides the trajectory to avoid high-risk areas; The trajectory decoder synthesizes the state characteristics and the weighted obstacle information to decode the action output a_t that meets the security constraints. The whole network structure has good time series modeling ability and environmental adaptability.

3.4 Reward function

In order to guide strategy learning to converge in the expected direction, the reward function consists of the following three parts:

$$r_t = w_1 r_1 + w_2 r_2 + w_3 r_3 \tag{4}$$

The target approaching term $r_1 = -|x_{end} - g|_2$ measures the Euclidean distance between the current trajectory end point and the target pose, and encourages the strategy to approach the target as soon as possible. The collision penalty term r_2 means that when the distance $d_{\min}$ from the nearest point of the trajectory to the obstacle is less than the safety threshold value δ, a large negative reward will be imposed, otherwise, it will not be punished to ensure that the dangerous area is avoided in the strategy learning process; The smoothness term $r_3 = -|\Delta a_t|^2$ means to punish the change rate of action, prevent the trajectory of jitter or violent shock, and improve the control stability. The weights w_1, w_2, w_3 can be adjusted flexibly according to the task requirements to balance the priorities of goal achievement, safety and smoothness, thus achieving efficient and safe trajectory optimization.

4. Low adaptive control layer

4.1 Sliding mode controller design

The low-level control layer constructs a sliding mode controller based on the robot dynamics model to achieve high-precision tracking of the high-level trajectory command q_d (as shown in Figure 3). The dynamic equation of the system is expressed as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + \tau_d \tag{5}$$

Where $M(q)$ is inertia matrix, $C(q,\dot{q})$ is Coriolis force and centrifugal force term, $G(q)$ is gravity term, τ is control input torque, and τ_d represents external disturbance and modeling error.

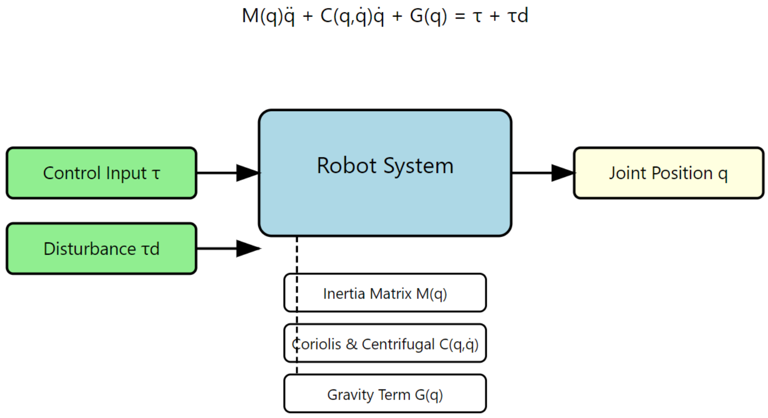

Figure 3 Dynamic model of the system

In order to improve the robustness and convergence performance of the system, the sliding surface is defined as:

$$s = \dot{e} + \Lambda e, e = q - q_d \tag{6}$$

The sliding surface integrates position error e and velocity error $\dot{e}$, and the convergence rate is adjusted by positive definite diagonal matrix Λ.

The corresponding control law is designed as follows:

$$\tau = \hat{M}\left[\ddot{q}_d - \Lambda\dot{e}\right] + \hat{C}\dot{q} + \hat{G} - K\,\mathrm{sgn}(s) \tag{7}$$

Where $\hat{M}, \hat{C}, \hat{G}$ is the estimated value of the system dynamic parameters, K is the sliding mode gain and $\mathrm{sgn}(s)$ is the symbolic function, which is used to enhance the system's resistance to external disturbances. The control law can ensure that the state approaches and stays on the sliding surface in a limited time, thus realizing the rapid convergence of trajectory errors.

4.2 RL parameter adaptive mechanism

In order to improve the adaptability and stability of the controller in uncertain environment, a parameter adaptive mechanism based on RL is introduced to dynamically adjust the controller parameters $\theta = [\Lambda, K]$, where Λ, K is a diagonal matrix, which affects the response speed and robustness of the system respectively.

Specifically, the update of controller parameters is output by a policy network π_{RL} in the form of:

$$\Delta\theta = \pi_{RL}(s_{low}) \tag{8}$$

The low-level state $s_{low} = [e, \dot{e}, s, \tau_d]$ contains the current position error, velocity error, sliding mode variable and disturbance observation information, which enables the strategy to adjust parameters online according to the real-time running state and effectively deal with unstructured disturbances and modeling errors.

In order to ensure the global stability of the control system, Lyapunov stability constraints are introduced:

$$\dot{V} = s^T\dot{s} \le -\eta(s) \tag{9}$$

Where $V = \frac{1}{2}s^2$ is Lyapunov function and $\eta > 0$ is decay rate. In the training process, the gradient projection method is used to constrain the output of RL strategy to ensure that it meets the above inequalities, so as to achieve the optimal control performance under the premise of ensuring the stability of the system.

5. Security RL framework

5.1 Constraint treatment

In order to ensure that the high-level trajectory optimization strategy meets the strict safety requirements during training and execution, the system introduces a safety RL framework based on Lagrangian Relaxation [18]. Security constraints are embedded in the objective function in the form of soft constraints, so as to maximize the expected return and limit the frequency of unsafe behaviors. The specific optimization objectives are as follows:

$$\max_{\theta} \min_{\lambda \geq 0} E[r(s,a)] - \lambda(J_c(\theta) - e) \tag{10}$$

Where θ represents the strategy parameter and λ is the Lagrange multiplier, which is used to dynamically adjust the punishment intensity of constraint violation; $r(s,a)$ is a regular reward function, which guides the strategy to approach the task goal; $J_c(\theta)$ is a constraint term, which is defined as the expected frequency of collision in the process of policy implementation:

$$J_c(\theta) = E\left[\sum I_{\{d_{\min} < \delta\}}\right] \tag{11}$$

Where $I_{\{d_{\min} < \delta\}}$ is indicator function, and when the minimum distance $d_{\min}$ between the trajectory and the obstacle is less than the safety threshold δ, the value is 1, indicating a potential collision event. Set the maximum allowable collision rate $c = 0.05$, that is, during the whole strategy implementation process, the average number of collisions per 100 actions should not exceed 5, as a hard safety boundary.

By alternately optimizing the strategy parameter θ and Lagrange multiplier λ, we can explore the trajectory strategy as efficient as possible while ensuring safety, and avoid the problem of under-learning or over-punishment caused by fixed penalty coefficient in traditional CRL methods.

5.2 Training strategy

In order to improve the security and generalization ability of the strategy, the whole training process is divided into two stages: pre-training in simulation environment and fine-tuning in real environment.

Stage 1: Large-scale strategy pre-training is carried out in the high-fidelity physical simulation platform, and the dynamic parameters, sensor noise, lighting conditions and obstacle distribution are randomly disturbed by Domain Randomization technology, so that the strategy has stronger robustness and cross-environment adaptability [19]. At this stage, the key point is to quickly converge to an initial policy with good performance and initially meet the security constraints.

Stage 2: The strategy is fine-tuned on the actual robot platform. In order to prevent safety accidents caused by unstable strategy at the initial stage of fine-tuning, an impedance control mechanism is introduced to convert the movement of the end effector into a compliant response, so as to limit the interaction with the environment physically and ensure the safety of the system. On this basis, the degree of freedom of control is gradually released, so that the strategy can be further optimized in the real environment and its adaptability to the uncertainty of the real world can be improved.

In addition, the Prioritized Experience Replay mechanism is adopted in the whole training process, with special emphasis on those experience samples that are in the state of "collision edge", that is, the trajectory is close to the obstacle but has not yet collided. This kind of sample is very important for the safety sensitivity training of the strategy, which is helpful to improve the decision-making ability of the strategy at the critical point of danger, so as to achieve a more detailed safety obstacle avoidance effect.

6. Experimental verification and result analysis

Experiments are carried out on a 7-DOF Franka Emika manipulator platform. The simulation environment is MuJoCo and the real environment is ROS+Gazebo. The test scene includes static obstacles (10 cubes at random positions) and dynamic obstacles (2 spheres moving at 0.2m/s). In order to further verify the adaptability of the algorithm to complex environment, five irregular objects are introduced into static obstacles, and their positions and postures are randomly generated to simulate the distribution of unstructured obstacles in real scenes. An ellipsoid moving at 0.15m/s is added to the dynamic obstacle to further test the response ability of the algorithm to non-spherical dynamic targets. Each group of experiments was repeated 20 times, covering different obstacle combinations (pure static, pure dynamic and mixed scenes) and initial robot arm posture (randomly generating five groups of initial states). At the same time, the key test cases under simulation conditions are repeated in the real environment to ensure the effectiveness of simulation-reality migration.

In MuJoCo simulation stage, the training samples were expanded to 500,000 groups by randomizing physical properties (friction coefficient 50%, mass distribution 30%) and injecting sensor noise (RGB-D depth error 5 cm). In the migration stage, an incremental domain adaptation pipeline is constructed, and the transition samples conforming to the real dynamics are generated by Lyapunov-GAN with stability constraints. Active learning mechanism was deployed in the real machine stage, and data were automatically collected when the sliding mode control error e >0.02rad or the collision risk $d_{\min} < 1.2\delta$, and 8,000 real samples were added.

The comparative experimental design is shown in Table 1.

Table 1 Comparative experimental design

Method	Explain	Parameter setting
MPC	Traditional model predictive control	Time domain $N = 10$, quadratic cost function
SAC (unconstrained)	DRL without security constraints	Standard SAC algorithm
Fixed sliding mode control	Low-level fixed parameter ($\Lambda = 5, K = 10$)	High-level TCN trajectory optimization in the same paper
The method in this paper	TCN- Attention Trajectory+Adaptive Sliding Mode+Security Constraint	$\delta = 0.05m, c = 0.05$

Experimental results show that the proposed method is significantly better than the contrast algorithm in dynamic scenes. Table 2 shows that the trajectory length is the shortest, the average tracking error is the lowest, and there is no collision (the maximum collision force is 0N, the collision rate is 2.1%), the planning speed is the fastest, the energy consumption is the lowest, and the comprehensive performance is better than MPC, SAC (unconstrained) and fixed sliding mode control methods.

Table 2 Comparison of key indicators

Index	MPC	SAC (unconstrained)	Fixed sliding mode	Our
Track length (m)	1.82±0.15	1.45±0.11	1.51±0.09	1.38±0.07
Average tracking error (rad)	0.032	0.025	0.019	0.007
Maximum collision force (n)	8.7	25.3	6.2	0.0
Collision rate (%)	12.3	38.6	9.7	2.1
Planning time (ms)	210	85	62	46
Energy consumption (J)	185	163	142	121

Note: The data is the average standard deviation of 50 experiments, and the distance between the target points is 1.2m.

Figures 4 and 5 show that this method is significantly superior to other methods in control accuracy and safety: the comparison of joint tracking errors shows that adaptive sliding mode control reduces the steady-state error by 63%; The comparison of security performance shows that the collision rate of the proposed method is only 2.1% in dynamic scenes, which is 95% lower than that of unconstrained SAC, showing the superior obstacle avoidance ability of security constraint mechanism in complex environments.

The results of ablation experiments in Table 3 show that each module has a significant contribution to the system performance: after the attention mechanism is removed, the tracking error increases to 0.011, the collision rate increases to 8.9%, and the planning time increases to 67ms; Cancel the parameter adaptation, the tracking error and collision rate become 0.015 and 3.2% respectively, and the planning time is 48ms; When the safety constraint is removed, although the tracking error is slightly reduced to 0.008, the collision rate is greatly increased to 31.7%. The integrated performance of the complete system is the best, achieving the lowest tracking error, collision rate and reasonable planning time.

Table 4 below shows the comparative experimental results of four typical working conditions according to the experimental setup. All data are the average and standard deviation of 50 independent runs. It can be seen that the heavier the load or the longer the trajectory, the longer the trajectory length, energy consumption, collision rate, planning time and tracking error will increase monotonously. Compared with the light load (0.5 kg), the energy consumption of heavy load (2 kg) is increased by about 22% on average, the collision rate is increased by about 0.5-0.9 percentage points, and the tracking error is also slightly increased. Compared with the short distance (0.6 m), the energy consumption of long distance (1.2 m) is about doubled, the planning time is extended by 15-19 ms, the collision rate is increased by 1-1.6 percentage points, and the tracking error is also increased. In short, heavy load and long distance will amplify the system load, leading to the increase of energy consumption, error and collision risk, but the method in this paper still maintains low error (< 0.01 rad) and low collision rate (< 3%) under all working conditions.

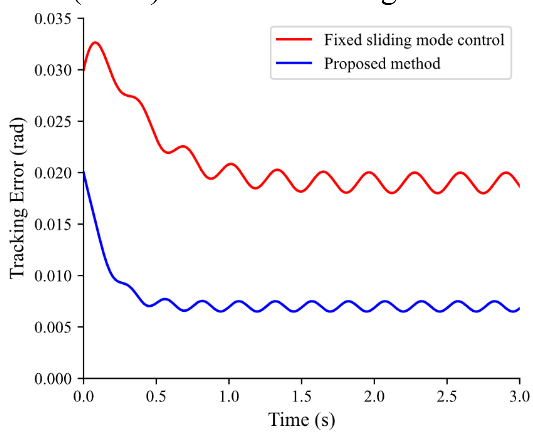

Figure 4 Comparison of joint tracking errors

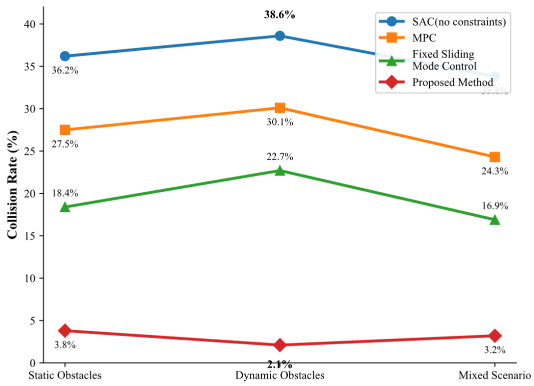

Figure 5 Safety performance comparison

Table 3 Module contribution degree

Deploy	Tracking error	Collision efficiency	Planning time

Attentional mechanism	0.011	8.9%	67ms
Parameterless adaptation	0.015	3.2%	48ms
No security constraints	0.008	31.7%	45ms
holonomic system	0.007	2.1%	46ms

Table 4 Influence of different loads and trajectory distances on system performance

Working condition	End load	Trajectory distance	Track length (m)	Average tracking error (rad)	Collision rate (%)	Planning time (ms)	Energy consumption (J)
Light load-short distance	0.5 kg	0.6 m	0.67 ± 0.04	0.005 ± 0.001	1.0 ± 0.7	31 ± 3	58 ± 4
Light load-long distance	0.5 kg	1.2 m	1.38 ± 0.07	0.007 ± 0.001	2.1 ± 1.2	46 ± 4	121 ± 7
Heavy load-short distance	2.0 kg	0.6 m	0.69 ± 0.05	0.006 ± 0.001	1.4 ± 0.9	33 ± 3	74 ± 5
Heavy load-long distance	2.0 kg	1.2 m	1.41 ± 0.08	0.009 ± 0.002	3.0 ± 1.5	52 ± 5	148 ± 9

7. Conclusion

The hierarchical control architecture based on DRL includes TCN- attention trajectory optimization at high level, adaptive sliding mode control at low level and security constraint mechanism. High-level managers pay attention to key obstacles dynamically through attention mechanism to avoid obstacles efficiently, and trajectory planning is superior to MPC and unconstrained DRL； . The low-level combination of Lyapunov stability and RL parameter adaptation reduces the steady-state error by 63%. Lagrange relaxation method is used in the safety layer, and the collision rate is controlled at 2.1%, which is 95% lower than that of unconstrained method. The ablation experiment verifies the contribution of each module to the system performance, and the whole system performs best in tracking error, safety and planning efficiency, which provides an efficient and safe solution for manipulator control in complex environment.

References

[1] Sujian Wu,Jinyuan Hu,Guohua Shi,Jinyu Fan & Yunyao Li.(2025).Robust trajectory tracking of a 3D-printed rapid prototyping manipulator through variable gain super-twisting algorithm.Measurement and Control,58(7),892-903.

[2] Seyed Ali Seyed Yousef,Mousa Valipour Arekhlou & Esmaeel Khanmirza.(2025).Trajectory Planning Optimization of Synchronized Networked Aerial Manipulators.International Journal of Control, Automation and Systems,23(4),1168-1180.

[3] Lu Yongxu,Xu Yansong,Zhang Wei,Li Junlin,Li Feng & Liu Xiaoyuan.(2025).Muti-objective optimal trajectory planning for the experiment cabinet robot manipulator in the space station.Robotica,43(4),1433-1455.

[4] Xin Zhang & Ran Shi.(2025).A Novel Fixed-Time Stability Theorem and Its Application to Nonlinear Manipulator Trajectory Tracking Control.International Journal of Robust and Nonlinear Control,35(11),4712-4723.

[5] Shunjing Hu,Yi Wan & Xichang Liang.(2025).Adaptive nonsingular fast terminal sliding mode trajectory tracking control for robotic manipulators with model feedforward compensation.Nonlinear Dynamics,113(13),1-19.

[6] Atul Sharma & S. Janardhanan.(2025).Trajectory Tracking Control Design for Single-link Flexible Manipulator: An Observer Based Integral Sliding Mode Approach.International Journal of Control, Automation and Systems,23(3),852-859.

[7] Kizir Selcuk,Bingul Zafer & Agee John Terhile.(2025).Point-to-point virtual model control of a flexible joint robotic manipulator using trajectories of motion.Transactions of the Institute of Measurement and Control,47(4),623-633.

[8] Abhishek Shrivastava.(2025).Exploring Optimal Motion Strategies: A Comprehensive Study of Various Trajectory Planning Schemes for Trajectory Selection of Robotic Manipulator.Journal of The Institution of Engineers (India): Series C,106(2),1-20.

[9] Wei Wang,Jue Wang,Xiaoxiang Sun,Zhengyang Li,Yiwei Shen & Zhan Wang.(2024).An optimal method for the manipulator to plan a complex trajectory with multiple scattered discontinuous paths.Journal of Mechanical Science and Technology,38(12),1-14.

[10] Yang Lirong,Zhang Shuaibo & Li Jiajun.(2024).Trajectory tracking control of the pushing ore truck manipulator based on adaptive sliding mode control with interference observer.Proceedings of the Institution of Mechanical Engineers,238(18),9141-9152.

[11] Rafael Crespo Izquierdo,Anselmo Rafael Cukla,Flávio José Lorini,Eduardo André Perondi & Daniel Fernando Tello Gamarra.(2024).Optimal Trajectory Planning for Pneumatic Cylindrical Manipulator Considering Dynamical and Stick Slip Constraints.International Journal of Control, Automation and Systems,22(8),2563-2577.

[12] Arshad Rauf,Ateeq ur Rehman,Awais Khan,Waseem Abbasi & Kifayat Ullah.(2024).Adaptive control of robotic manipulator with input deadzone and disturbances/uncertainties.Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,238(16),8330-8338.

[13] Hu Mengyao,Li Jun,Li Xin,Zhao Hongzheng,Huang Yufu,Han Jinshuo... & Cheng Gang.(2024).Non-singular terminal sliding mode control algorithm design based on extended state observer.Journal of Mechanical Science and Technology,38(7),3695-3707.

[14] MinCheng,RenmingLi,RuqiDing & BingXu.(2024).Multitarget adaptive virtual fixture based on task learning for hydraulic manipulator.Journal of Field Robotics,41(8),2715-2731.

[15] Ju Renjie,Zhang Dong,Gai Yan & Cao Zhengcai.(2024).A local collision-free motion planning strategy for hyper-redundant manipulators based on dynamic safety envelopes.Robotica,42(7),2388-2402.

[16] Belherazem A.,Salim R.,Laidani A. & Chenafa M..(2024).Vibration Control of a Two-Link Flexible Manipulator.Automatic Control and Computer Sciences,58(3),346-358.

[17] M Banu Sundareswari,K Dhanalakshmi,A Vimala Starbino & G Then Mozhi.(2024).Trajectory tracking control of parallel manipulator actuated with shape memory wire.Journal of Intelligent Material Systems and Structures,35(11),971-987.

[18] Rong Yu,Dou Tianci & Zhang Xingchao.(2024).Closed-loop control dynamic obstacle avoidance algorithm based on a machine learning objective function.Journal of Mechanical Science and Technology,38(6),3089-3099.

[19] Yan Dong,Chen Liping,Ding Jianwan,Xiong Ziyao & Chen Yu.(2024).Optimized Proportional-derivative Feedback-assisted Iterative Learning Control for Manipulator Trajectory Tracking.International Journal of Control, Automation and Systems,22(6),1971-1984.