Research on Optimization of Robotic Motion Control Algorithms based on Deep Reinforcement Learning

Ke Yang*
Engineering College
Beijing Forestry University
Beijing, China
*1316175231@qq.com

***Abstract*—It has become the key problem solution in the field of intelligent robot to obtain high-precision and high-efficient action decision-making ability under dynamic and complex environment in the robot motion control of high-dimensional continuous action space. Built upon the existing framework of the Soft Actor-Critic (SAC) algorithm, this paper innovatively incorporates multi-channel sensor data fusion and hierarchical strategy planning: firstly, multi-layer convolutional network and attention mechanism are employed to extract and compact representative features of multi-source signals (cameras, force sensors, lidars, etc.). Next, the adaptive reward module and the target prediction unit (LSTM) were seamlessly embedded within the actor network of SAC, which can dynamically adapt the estimation of action value based on historical state and uncertainty during training. Finally, a hierarchical strategy planning structure is designed to combined the global search strategy and local fine decision making strategy to achieve the fine control and long-time sequence planning of robot movement, so as to ensure the stable performance under the conditions of high-speed movement and complex obstacle environment. The experimental results demonstrate that in a variety dynamic scenarios, the model convergence speed and control accuracy are significantly superior to the traditional SAC and the other comparison methods.**

Keywords: Robot motion control, Deep reinforcement learning, SAC algorithm, Hierarchical strategy.

I. INTRODUCTION

One of the core issue in robotics is robot motion control, which also is the direct influence factors of robot autonomy and adaptive capacity. An efficient and robust motion control system not only decides the survivability of the robot in complex environments, but also directly affects the quality of task, and its working efficiency. Traditional motion control methods are highly dependent on precise mathematical modeling and sophisticated solution techniques for trajectory planning, including PID control, optimal control, and model predictive control (MPC) using kinematic or dynamic equations [1].

These methods succeed in a structured atmosphere ensuring stability and movement predictable. Still, they tend to be limited in complex or non-stationary environments (e.g., in unstructured terrain, human-computer interaction, and in response to untimely bursts) and fail to capture dynamic changes promptly [2]. This is mainly attributed to the high uncertainty in the external environment as well as the high variability of mission objectives in the real world, which make accurate modeling, an essential component of the traditional control method, tedious, and entropistic, making it close to impossible.

Reinforcement learning is a major branch of machine learning, and agents can continually learn external environment optimal strategy by interacting with environment in a trial and error process so as to maximize the cumulative reward. Reinforcement learning doesn't need an accurate system model like traditional control methods; it learns data-driven which is an optimal motion control strategy. This property suggests that reinforcement learning has great potential in solving complex, high dimensional, and nonlinear system control problems [3].

However, early-stage reinforcement learning methods (including Q-learning and policy gradient methods) suffer from low computing efficiency, slow convergence rate and poor sample efficiency among continuous states and actions space. For instance, value-based approach has trouble in handling the high-dimensional state space, and the strategy-based gradient-based approach is easily stuck in the local optimum [4]. Moreover, the use of a large amount of interaction data for training is also an important characteristic of reinforcement learning, but for robot motion control tasks, the robot needs to go through thousands of attempts and failures in order to learn a stable motion strategy, which is almost impossible to achieve in practical applications.

Deep learning is powerful approach for working with high dimensional data and now days deep learning is rising. Deep Reinforcement Learning (DRL) emerged by combining deep learning with reinforcement learning. In this paradigm, neural networks discover promising control strategies straight from high-dimensional input data (cameras, lidar, etc.) by selecting the right features and associating them with the corresponding control trajectory using reinforcement learning. Compared with conventional reinforcement learning, deep reinforcement learning (DRL) can effectively process complex continuous states and action spaces, which enhances the adaptive capacity of robots operating in dynamic environments [5]. One well-known success is the application of Deep Q Network (DQN) to complex game environments, and other methods such as Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimization (PPO) have also advanced the capabilities of DRL applied to continuous control tasks, allowing robots to learn complex motion patterns including balance control, dynamic obstacle avoidance, and multi-joint coordinated movement [6].

However, although DRLs have performed outstanding results in the simulation environment, there still exist numerous challenges in practical applications. Sample inefficiency is

among them, a significant issue. Training millions of times in a virtual environment is quite cheap, whereas the training cost of a real robotic system is prohibitively high as one training will provide hardware wear and tear, energy expenditure besides the risk of damage. Moreover, the training process of DRL is usually black-box, i.e., the learning process of the policy has no clear explainability, which results in uncertain during its deployment. For example, in a complex environment a neural network may learn some locally optimal strategies that are either not expensive or not generalizable, consequently leading to bad robot behavior outside of a specific scenario. Moreover, the heavy computational burden of DRL model also prevent its application in some embedded systems.

II. Related Work

Nonoyama et al. [7] set objectives for minimizing the robot's execution time and overall energy. On the first step, they implemented initial tuning of the PID controller that was followed by refinement of the PID parameters by means of meta-heuristic algorithms including genetic algorithm (GA) and particle swarm optimization (PSO) to create more precise robot movement trajectories for the purpose of implementing energy-saving robot design. NEO (a fast, purely reactive motion controller allowing a manipulator to navigate around static and dynamic environments while approaching a desired end-effector position), was proposed by Haviland and Corke [8] allowing the manoeuvrability of the robot to be maximised while preserving smooth and safe movement by asymptotically approaching the limits on joint position and speed. NEO builds a strict convex quadratic programming problem genrally solved in milliseconds including obstacles, joints limits and maneuverability that can applied to robots with 7 DOF.

Štefek et al. [9] considered all the devising methodology of course for tuning fuzzy logic controllers for differential drive wheeled mobile robots. They also use meta-heuristic algorithms like genetic algorithm and particle swarm optimization to optimize the parameters of fuzzy logic controller to achieve a more accurate robot motion trajectory. The method enhances the ability of the robot to follow the given trajectory and the stability of the motion, and verifies in complex environments.

Wensing et al. [10] investigated the use of model-based optimization algorithms for real-time motion planning and control of dynamically stable legged robots. While noting that solving the most general optimal control problems remains difficult in real-time online computing, they also emphasize that these problems become computationally feasible through suitable model simplification and environmental exposure handling. Israr et al. [11] funneled through several optimization techniques that apply to UAV motion planning. “In this paper, some of the many scientific papers related to biomimetic algorithms used in motion planning.

Liu et al. [12] Genetic Algorithms-Based Trajectory Optimization for Mobile Robots Using Robot Digital Twin Technology. They will conduct a study on motion trajectory planning to enhance the accuracy of the robot and address some of the critical issues affecting the work efficiency of the robot. Awwal et al. [13] proposed a novel spectral RMIL+ conjugate gradient method to deal with unconstrained optimization problems and applied it to the fields of portfolio selection and robotic motion control. In the area of optimization algorithms, their methodology has been proven efficient for addressing robotic motion control problems in a novel way.

III. Methodologies

A. Multi-Channel Data Fusion and Soft Actor-Critic

In scenarios where robots perform high-precision and high-real-time motion control, it is often necessary to obtain multi-modal information about the external environment and its own state from multiple channels such as cameras, force sensors, and lidars. To do this, we first perform feature extraction on the raw data $\{x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(M)}\}$ of each channel through a parallel multi-layer convolutional network (CNN) to obtain a set of intermediate feature representations $\{h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(M)}\}$. Let $f_\psi^{(m)}(\cdot)$ denote the m-th convolutional network, as Equation 1:

$$h_t^{(m)} = f_\psi^{(m)}\left(x_t^{(m)}\right), m = 1, \dots, M. \tag{1}$$

In order to integrate these multi-source features and focus on the most critical part of decision-making, attention mechanism is introduced. Considering the feature pile $H_t = \{h_t^{(1)}, h_t^{(2)}, \dots, h_t^{(M)}\}$ as a key-value pair (K, V), given the query vector Q (generated by the trainable vector or the global state), attention is calculated as Equation 2:

$$Attention(Q, K, V) = softmax\left(\frac{QK^\top}{\sqrt{d}}\right)V, \tag{2}$$

where d is the scaling factor. Re-mapped $g_\omega(\cdot)$ Normalize or reduce the dimension to obtain the final fusion representation z_t, as shown in Equation 3:

$$z_t = g_\omega\left(Attention(Q, K, V)\right). \tag{3}$$

On this basis, the multi-source fusion feature z_t and the environment state s_t were jointly input into the SAC (Soft Actor-Critic) framework. The Soft Q-Function of the SAC core can be written in Equations 4 and 5:

$$Q_\theta(s_t, a_t) \approx r(s_t, a_t) + \gamma \mathbb{E}[V_{\bar{\theta}}(s_{t+1})], \tag{4}$$

$$V_{\bar{\theta}}(s_{t+1}) = \mathbb{E}_{a_{t+1}\sim\pi_\theta}\left[Q_\theta(s_{t+1}, a_{t+1}) - \alpha log \pi_\phi(a_{t+1}|s_{t+1})\right], \tag{5}$$

where π_ϕ is the routing network, α entropy coefficient, θ and $\bar{\theta}$ are the parameters of the online Q network and the target Q network, respectively. In order to take into account the environmental uncertainty and task requirements, the adaptive reward term $\Delta r(s_t, a_t)$ is introduced on the basis of the original reward $r(s_t, a_t)$ to obtain the updated reward, as shown in Equation 6:

$$r'(s_t, a_t) = r(s_t, a_t) + \lambda\, \Delta r(s_t, a_t), \tag{6}$$

where λ is the balance parameter, and Δr can be adjusted adaptively according to a variety of real-time feedbacks such as collision risk, speed demand, etc.

At the same time, in order to enhance the ability to model future states or reward sequences, LSTM units are embedded in the Actor or Critic bypass to memorize the historical sequence $\{s_\gamma, a_\tau\}$ and predict the next moment state or reward correction, as shown in Equations 7 and 8:

$$h_{t+1}^{(LSTM)} = LSTM\left(h_t^{(LSTM)}, [s_t, a_t]\right), \tag{7}$$

$$\hat{r}'_{t+1} = f_\xi\left(h_t^{(LSTM)}\right), \tag{8}$$

where $\hat{r}'_{t+1}$ is the predicted reward correction at the next moment, which is used to dynamically correct the learning direction of the strategy or value function in a highly uncertain environment. If the prediction error $\delta_t = \| \hat{r}'_{t+1} - r'_{t+1} \|^2$ is regarded as a measure of environmental uncertainty, it can be further fed back into the reward or entropy coefficient adjustment to improve the stability and robustness of the model.

B. Hierarchical Strategic Planning

In order to take into account both the global goal and the local fine control in the high-speed and complex obstacle scenarios, a hierarchical strategy planning structure is introduced into the SAC framework, and the global strategy π_ϕ^G and the partial strategy π_η^L are defined respectively. At time step t, the global strategy starts from the fusion feature z_t and the state s_t to generate the sub-target $g_t \sim \pi_\phi^G(g|s_t)$. The local strategy is based on (s_t, g_t) to produce specific actions $a_t \sim \pi_\eta^L(a|s_t, g_t)$. When the two-level strategy is implemented in tandem, the global level is responsible for macro decision-making and sub-goal selection, while the local level focuses on real-time obstacle avoidance and motion refinement to meet the multiple needs of efficiency and safety in complex scenarios. During training, each layer can use the SAC idea to maintain the value function and add entropy regular terms. For example, the Q function corresponding to a local strategy is as Equations 9 and 10:

$$Q_\varphi^L(s_t, g_t, a_t) \approx r^L(s_t, a_t) + \gamma \mathbb{E}\left[V_\varphi^L(s_{t+1}, g_t)\right], \tag{9}$$

$$V_\varphi^L(s_{t+1}, g_t) = \mathbb{E}_{a_{t+1} \sim \pi_\eta^L} \left[Q_\varphi^L(s_{t+1}, g_t, a_{t+1}) - \alpha^L log \pi_\eta^L(a_{t+1}|s_{t+1}, g_t)\right]. \tag{10}$$

The global strategy gets the corresponding reward $r^G(s_t, a_t)$ in the process of generating the sub-goal, and combines the optimization of entropy to form the global value function. In actual deployment, the global strategy is responsible for task-level or path-level planning, and the local strategy completes rapid obstacle avoidance and trajectory tracking, so as to achieve stable, accurate and efficient action decision-making in high-speed motion and complex obstacle environments.

After integrating multi-source data fusion, attention mechanism, adaptive reward and LSTM prediction into the above-mentioned hierarchical SAC structure, the robot can adaptively balance exploration and utilization in different scenarios, while taking into account the global and local control objectives. It reflects the superiority of multi-channel sensor fusion and hierarchical strategy in actual robot motion control.

Our model leverages multi-channel data fusion, which combines cameras, lidar, and force sensors for a comprehensive environmental understanding. The attention mechanism then enhances this fusion by assigning weights to the most critical features, improving decision-making during high-dimensional motion planning.

IV. Experiments

A. Experimental Setup

For this study, we applied a deep reinforcement learning-based robot motion control algorithm in the KITTI dataset (http://www.cvlibs.net/datasets/kitti/) for the training and evaluation of this paper. Based on high-resolution images with multiple modalities of driving scenarios, LiDAR point cloud, GPS positioning and IMU inertial measurements collected jointly by the Karlsruhe Institute of Technology and the Toyota Institute of Technology, the dataset has become a well-known driving challenge in the research fields of autonomous driving and robotics. This study uses the real-world environment perception data given by the KITTI dataset to train robot movement control strategies on complex road environments, and evaluate their performances on the same data to verify the validity and robustness of the algorithm.

To compare the performance of the deep reinforcement learning-based robot motion control algorithm, four comparison methods were selected.

- Initially, the Proportional-Integral-Derivative Controller (PID) design is a classic feedback control approaches to guarantee stable tracking for the trajectory under the regulation of proportional (P), integral (I), and differential (D) parameters.
- Similarly, imitation learning (IL) enables robots to gather knowledge from the expert-provided data so as to achieve a functional motion control strategy, i.e., an appropriate way of acting for environments in which explicit tasks are present, but no reward signal is available.
- Furthermore, Model Predictive Control (MPC) is utilized to predict the system state based on the system dynamics model and optimize the control inputs at each control cycle to improve the stability and real-time performance of robot motion.
- Finally, classical reinforcement learning (RL) allows agents to learn the best strategies through exploration and exploitation with the environment.

B. Experimental Analysis

The success rate is defined as the percentage of successful task completions within a given number of training episodes. As shown in Figure 1, although the traditional PID and IL methods are relatively stable in the initial stage, their success rate is slow and the overall level is low.

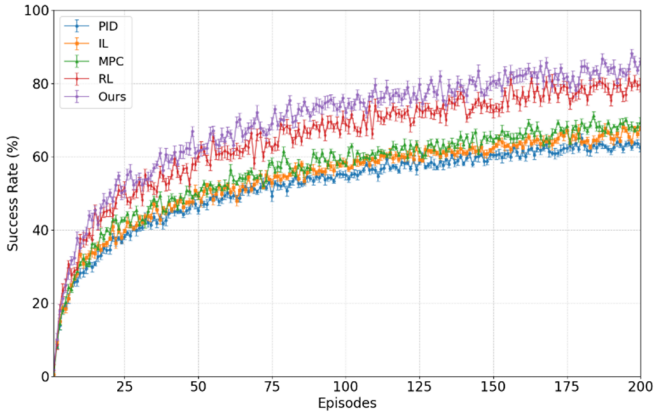

Figure 1. Comparison of Success Rates Across Different Methods.

MPC and conventional RL methods enhance the success rate tentatively but unable to maintain the stability as they show a big fluctuation in complex dynamic environments. In contrary, our proposed method may have slow growth at the beginning, but the success rate of our method rises sharply with the training deepening and has a huge advantage in the later stage with a high stability and robustness. Our method exhibited superior ability in resolving high-dimensional continuous motion control and complex obstacle environment with a wide and intuitive application of multi-channel data fusion, attention mechanism, and hierarchical strategy planning.

The trajectory length represents the total distance traveled by the robot from the starting point to the goal. As the obstacle density increases in Figure 2, the average trajectory length of each method increases, but different methods have different degrees of increase.

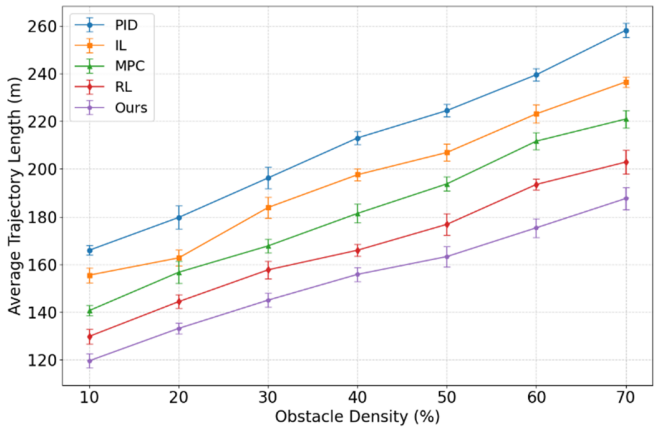

Figure 2. Comparison of Average Trajectory Length.

Path smoothness measures how smooth the robot's path is, quantified by the number of sharp turns or deviations in the trajectory. It can be clearly seen that the traditional PID and IL methods have large amounts of path redundancy and rapid increases in trajectory length in complex environments, indicating their low planning efficiency in high-density obstacle conditions. While the MPC and conventional RL approaches have improved, they still exhibit significant variances in high-density regions.

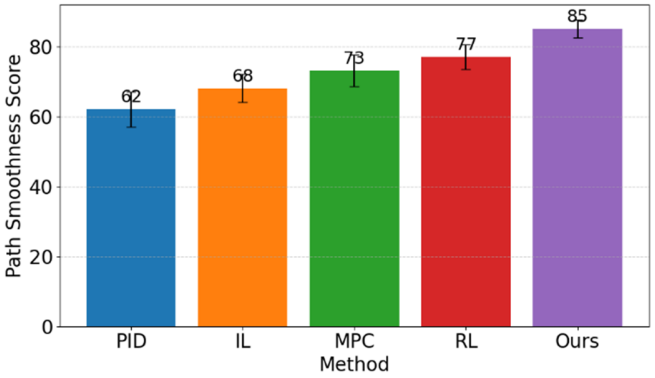

Figure 3. Comparison of Path Smoothness.

Path smoothness is a metric representing the differences in performance of the methods shown in Figure 3. Latency in path planning is reflected in the smoothness scores for the path, observed to be low for both PID and IL methods, thus, the path consists of many jitters and sharp turns. Finally, although the smoothing of the MPC and conventional RL methods have been improved, it is still unsatisfactory. Whereas our approach exhibits better performance on path smoothness and obtains the highest score, demonstrating the efficiency of our method in suppressing excessive steering and oscillation in the path planning process and yielding a smoother and more smooth motion trajectory.

V. Conclusions

In conclusion, this study innovatively integrates multi-channel sensor data and hierarchical strategy planning, and introduces an adaptive reward module and a target prediction unit (LSTM) to improve the motion control performance of the robot in a complex dynamic environment. Experimental results show that our method is superior to the traditional PID, IL, MPC and conventional RL methods in key indicators. In the future, we will explore the problem of policy generalization in high-dimensional environments, combining feedback mechanisms

References

[1] Kashyap, Abhishek Kumar, and Dayal R. Parhi. "Optimization of stability of humanoid robot NAO using ant colony optimization tuned MPC controller for uneven path." Soft Computing 25.7 (2021): 5131-5150.

[2] Zhang, Mingyang, and Jihong Yan. "A data-driven method for optimizing the energy consumption of industrial robots." Journal of cleaner production 285 (2021): 124862.

[3] Lee, HyeokSoo, and Jongpil Jeong. "Mobile robot path optimization technique based on reinforcement learning algorithm in warehouse environment." Applied sciences 11.3 (2021): 1209.

[4] Elsisi, Mahmoud, et al. "Improvement of trajectory tracking by robot manipulator based on a new co-operative optimization algorithm." Mathematics 9.24 (2021): 3231.

[5] Liao, Zhao-Yang, et al. "Optimization of robot posture and workpiece setup in robotic milling with stiffness threshold." IEEE/ASME transactions on mechatronics 27.1 (2021): 582-593.

[6] Rafai, Anis Naema Atiyah, Noraziah Adzhar, and Nor Izzati Jaini. "A review on path planning and obstacle avoidance algorithms for autonomous mobile robots." Journal of Robotics 2022.1 (2022): 2538220.

[7] Nonoyama, Kazuki, et al. "Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization." Energies 15.6 (2022): 2074.

[8] Haviland, Jesse, and Peter Corke. "NEO: A novel expeditious optimisation algorithm for reactive motion control of manipulators." IEEE Robotics and Automation Letters 6.2 (2021): 1043-1050.

[9] Štefek, Alexandr, et al. "Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot." Applied Sciences 11.13 (2021): 6023.

[10] Wensing, Patrick M., et al. "Optimization-based control for dynamic legged robots." IEEE Transactions on Robotics 40 (2023): 43-63.

[11] Israr, Amber, et al. "Optimization methods applied to motion planning of unmanned aerial vehicles: A review." Drones 6.5 (2022): 126.

[12] Liu, Xin, et al. "Genetic algorithm-based trajectory optimization for digital twin robots." Frontiers in Bioengineering and Biotechnology 9 (2022): 793782.

[13] Awwal, Aliyu Muhammed, et al. "A spectral RMIL+ conjugate gradient method for unconstrained optimization with applications in portfolio selection and motion control." IEEE Access 9 (2021): 75398-75414.