文章编号:1671-4598(2025)07-0171-09 DOI:10.16526/j.cnki.11-4762/tp.2025.07.021 中图分类号:TP242 文献标识码:A

基于 DDPG-PID 控制算法的机器人高精度运动控制研究

赵坤灿[1]，朱 荣[2]
(1. 昆明冶金高等专科学校 教务处，昆明 650033；
2. 昆明理工大学城市学院 信息工程系，昆明 650500)

摘要：随着工业自动化、物流搬运和医疗辅助等领域对机器人控制精度要求的提高，确保运动控制的精确性成为关键；对四轮机器人高精度运动控制进行了研究，采用立即回报优先机制和时间差误差优先机制优化深度确定性策略梯度算法；并设计了一种含有两个比例－积分－微分控制器的高精度系统；在搭建底盘运动学模型的基础上，分别为 x、y 方向设计了独立的 PID 控制器，并利用优化算法自适应地调整控制器的参数；经实验测试 x 向上优化算法控制的跟踪误差为 0.097 6 m，相较于优化前的算法误差降低了 9.76%；y 向上优化算法的跟踪误差为 0.108 8 m，优化算法误差较比例－积分－微分控制器减少约 48.0%；经设计的控制系统实际应用满足了机器人运动控制工程上的应用，稳态误差和动态误差分别为 0.02 和 0.05；系统误差较小，控制精度高，适合精细控制任务，为机器人高精度运动控制领域提供了新的技术思路。

关键词：机器人；PID；DDPG；精度；控制系统

High-precision Motion Control of Robots Based on DDPG-PID Control Algorithm

ZHAO Kuncan[1], ZHU Rong[2]
(1. Academic Affairs Office, Kunming Metallurgy College, Kunming 650033, China;
2. Department of Information Engineering, City College, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: With the increasing demand for robot control accuracy in fields such as industrial automation, logistics handling, and medical assistance, it is crucial to ensure the precision of motion control; This paper presents a high-precision motion control method for a four-wheel robot, optimizes a depth deterministic strategy gradient algorithm using an immediate reward priority mechanism and a time difference error priority mechanism, and innovately designs high-precision system with two proportional-integral-derivative (PID) controllers. On the basis of building a chassis kinematic model, independent PID controllers are designed for the x and y directions respectively, and optimization algorithms are used to adaptively adjust the parameters of the controllers; Through experimental testing, the tracking error controlled by the x-direction optimization algorithm is 0.097 6 m, which is 9.76% lower than that of the algorithm before optimization; The tracking error of the y-direction optimization algorithm is 0.108 8 m, and the optimization algorithm reduces the error by about 48.0% compared with the PID controller; The designed control system meets the practical application of robot motion control engineering, with a static error and dynamic error of 0.02 and 0.05, respectively; The system has the characteristics of small error, high control accuracy, and suitability for fine control tasks, providing new technical ideas for high-precision motion control of robots.

Keywords: robots; PID; deep deterministic policy gradient (DDPG); accuracy; control system

收稿日期:2025-02-20； **修回日期**:2025-04-01。

作者简介:赵坤灿(1982-),男,硕士,副教授。
朱 荣(1965-),男,硕士,教授。

引用格式:赵坤灿,朱 荣.基于 DDPG-PID 控制算法的机器人高精度运动控制研究[J].计算机测量与控制,2025,33(7):171-179.

0 引言

高精度运动控制在机器人技术中至关重要，其不仅关系到任务的完成质量，还直接影响机器人的可靠性和稳定性，是机器人技术发展的关键领域之一。四轮机器人具有较高的稳定性和易控制的结构，其在工业控制、物流运输、自动驾驶等领域被广泛应用。文献［1］针对四轮机器人在室内外的狭窄空间的运动控制，设计了一种旋转机械齿轮系驱动的小轮子，结果表明该方法具有抗倾覆和防滑稳定性。尽管近年来机器人控制技术取得了显著进展，但在实际应用中仍面临着诸多挑战。比如现有的控制算法，比例－积分－微分（PID，proportional integral derivative）控制，虽然在工业控制领域得到了广泛应用，但其在面对复杂环境、动态任务和非线性系统时，其参数调整难度较大，难以适应复杂变化的控制需求。为了提高机器人操作精度和可控性，文献［2］提出结合 PID 技术与模糊控制器来优化机器操作系统，验证了其一定的有效性。近年来，深度强化学习因其结合了深度学习的感知能力和强化学习的决策能力，能够解决传统控制方法难以处理的复杂、高维、非线性问题。深度确定性策略梯度（DDPG，deep deterministic policy gradient）作为强化学习中的一种算法，凭借其处理连续动作空间的能力，逐渐成为强化学习中的研究热点。文献［3］基于 DDPG 算法设计了针对机器人旋转倒立摆系统的控制方法，实验结果表明该方法相较于传统 PID 控制效果更佳。尽管 DDPG 适用于解决连续动作空间的问题，并通过目标网络和经验回放技巧提升算法稳定性和收敛速度，但传统 DDPG 算法存在奖励函数曲线不稳定、训练收敛速度较慢等局限性，这使得其在一些应用场景中未能完全满足高精度和高实时性的需求[4-5]。因此，研究创新性地采用双机制优化的 DDPG 算法，并将其应用于高精度双 PID 控制系统中。研究旨在提高运动控制的精度，实现机器人性能提升，为机器人在复杂和变化的环境下保持较高的控制精度提供了有效的技术方案。

1 引入双机制的 DDPG 算法优化设计

1.1 立即回报优先机制和时间差误差优先机制

在强化学习中，智能体在每一步决策后会获得一个立即回报，立即回报优先机制依据立即奖励的大小来选择采样样本，确保高回报的样本优先被采样和学习[6]。优先选择高回报的样本进行训练，确保在训练过程中，智能体能够聚焦于那些能够带来较大回报的行为策略。这种方式可以减少无效样本的采样，提高训练的效率。通过将这一机制引入 DDPG 算法，智能体每次训练时都会根据当前状态和动作计算获得立即的奖励，并用其作为选择下一步训练样本的依据，优化了训练过程。具体而言，t 时刻环境与强化学习的智能体交互，此时智能体状态为 s_t，在策略 π 下以动作 a_t 进入下一次训练，从而获得 $t+1$ 时刻的状态 s_{t+1}，同时获得立即回报 r_t。每个经验样本的奖励优先级 P_r 可以根据 r_t 来计算，P_r 计算式如式（1）所示：

$$P_r = r(s_t, a_t) \tag{1}$$

式（1）说明该机制优先选择那些获得高立即回报的经验进行采样和学习，以确保智能体集中训练那些带来较高奖励的行为策略。在时间差（TD，time difference）误差优先机制中，智能体会根据 TD 误差的大小来决定哪些样本具有更高的学习价值。误差越大的样本意味着当前模型的预测存在更大的偏差，因而这些样本被赋予更高的采样优先级[7]。通过引入此机制，算法可以更集中地训练于那些预测误差较大的样本，加速模型的优化过程。立即回报优先机制和 TD 误差优先机制的设计流程，如图 1 所示。

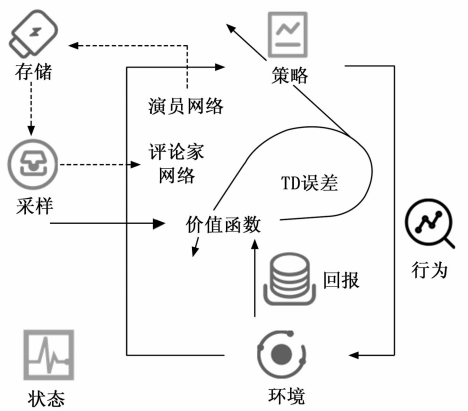


图 1 双机制设计流程示意图

图 1 中，演员网络选择智能体的动作，而评论家网络评估所选的动作。结合 TD 误差，评论家网络能够评估采取策略下的每一个输出动作的价值。其中采样的优先级依据 TD 误差来决定，TD 误差越大，说明当前模型的预测精度还有很大的提升空间，因此这些样本应该被优先学习和更新[8]。TD 误差是指当前预测值与目标值之间的差异，反映了智能体当前模型的预测误差。在强化学习中，评论家负责评估行为的质量，并生成 δ_t，对 Q_t 进行评价。这一机制使得学习过程更加高效，能够快速反馈并调整策略。具体而言，TD 误差 δ_t 即 $t+1$ 时刻和 t 时刻的 Q 值函数之间的差值，计算式如式（2）所示：

$$\delta_t = \gamma Q^{\pi}(s_{t+1}, a_{t+1}) + r_{t+1} - Q^{\pi}(s_t, a_t) \tag{2}$$

式（2）中，$\gamma Q^{\pi}(s_{t+1},a_{t+1})+r_{t+1}$ 为 $t+1$ 时刻 π 策略下的状态 s_{t+1}，采取动作 a_{t+1} 的 Q 值目标函数，γ 为权重系数。$Q^{\pi}(s_t,a_t)$ 为 t 时刻相同策略下的 Q 值目标函数。通过记录每个样本的 TD 误差，并按照误差的大小来决定样本的回放顺序，可以确保那些预测误差较大的样本被更频繁地使用，从而提高学习的效率[9]。该机制需要在每次更新过程中，计算每个样本的 TD 误差。将 TD 误差存储在经验回放池中，并更新每个样本的优先级。在训练过程中，优先选择 TD 误差较大的样本进行回放和更新，直到其 TD 误差减小到一定程度。随着模型的改进，TD 误差会逐渐减小，因此需要动态调整样本的优先级，确保模型能够持续学习到新的、未被充分学习的样本[10]。

1.2　DDPG 算法优化设计

结合立即回报优先机制和 TD 误差优先机制的优势，同时考虑到在传统的 DDPG 算法中，常常存在奖励函数曲线不稳定、训练收敛速度慢等问题，限制了其在复杂环境下的表现和应用[11]。为了提高 DDPG 算法的训练效率和精度，研究选择双机制对 DDPG 算法进行优化。优化 DDPG 算法流程示意图，如图 2 所示。

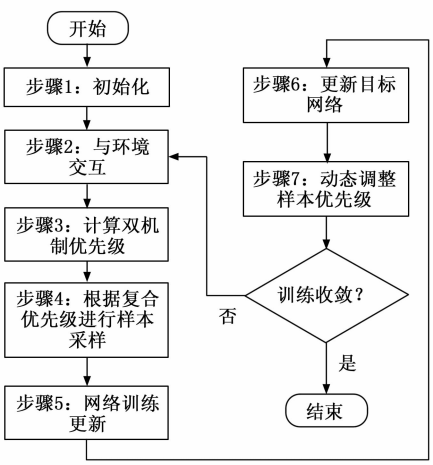


图 2　优化 DDPG 算法流程示意图

图 2 中，优化 DDPG 算法步骤 1 为初始化演员网络和评论家网络及其对应的目标网络，以及经验回放池。步骤 2 为与环境交互，包含智能体在环境中根据当前策略网络选择动作，执行动作后，获得下一状态和奖励，形成经验元组。之后，将经验元组存入经验回放池。步骤 3 为分别计算两种机制的优先级，然后按比例合并二者，研究设定立即回报、TD 误差的经验优先级分别为 P_i、P_j，这两个机制的经验优先级的计算式如式（3）所示：

$$P_i=r_t+\varepsilon,P_j=|\delta_t|+\varepsilon \tag{3}$$

式（3）中，ε 为引入的常数，用于保证每一个经验优先级不为 0。在经验回放池中，TD 误差越大的样本被优先采样和回放，直到这些误差减小到一定程度[12]。每次训练更新过程中，算法通过计算每个样本的 TD 误差，并动态调整样本的优先级。步骤 4 为通过复合优先级 u_k 的排序方法对样本进行采样，保证模型能持续学习那些未被充分学习的重要样本。结合立即回报优先机制与 TD 误差优先机制，计算 u_k。u_k 的计算还结合 P_i、P_j 这两个优先级的比重，将经验样本由大到小依次进行排序，获得 $rank(i)$、$rank(j)$。根据一定比例，重新排序得到 u_k 如式（4）所示：

$$u_k=\alpha rank(i)+\beta rank(j) \tag{4}$$

式（4）中，α、β 均为比例数，有 $\alpha+\beta=1$。为了保证两者权重相等，α、β 均为 0.5[13]。通过设置一定比例的重新排序，得到最终的优先级排序，样本的采样概率与优先级成正比[14]。设经验池中有 n 个经验，重新排序的优先级 P_k 和经验采样的概率 G_k 如式（5）所示：

$$P_k=\left(\frac{1}{u(k)}\right)^{\eta},G_k=\frac{P_k}{\sum_n P_n} \tag{5}$$

式（5）中，η 为系统采样时优先级程度，$0\leqslant\eta\leqslant1$。若 η 为 0，则 P_k 为 1，即均匀采样。因此，系统能够结合 TD 误差与回报机制，使得 DDPG 算法在面对复杂环境时，能够更加高效、准确地进行训练和决策。研究在这两种机制的作用下，将相同的经验在两种排序方法下以一定比例进行复合排序，对该经验样本的复合优先级进行计算，以此计算经验采样概率。而这个概率能够有助于在经验池中选择高优先级的经验。双机制的设计可以优先对更为重要的经验进行重复采样，使得算法收敛速度加快，并降低了局部最优造成的误差。研究通过动态调整优先级和采样策略，帮助智能体学习到更多有用的信息，提高预测精度。步骤 5 为网络训练更新，在优化过程中，智能体的目标是最小化损失函数，通过更新网络的参数来降低 TD 误差。将损失函数 L 定义为当前 Q 值与目标 Q 值之间的差异平方，L 计算式如式（6）所示：

$$L=E_{all}\times\{[\gamma Q^{\pi}(s_{t+1},a_{t+1})+r_{t+1}-Q^{\pi}(s_t,a_t)^2]\} \tag{6}$$

式（6）中，E_{all} 为所有样本的期望。步骤 6 为使用软更新方式更新目标网络参数 M，如式（6）所示：

$$M=\frac{1}{K}\sum_{t=1}^{K}\frac{dQ(s_t,a_t)}{da_t}\cdot\frac{\mathrm{d}a_t}{\mathrm{d}\vartheta} \tag{6}$$

式（6）中，$k\in K$ 表示批量样本的大小。ϑ 为参数化策略函数。步骤 7 为动态调整样本优先级，具体来说，根据网络训练结果，重新计算各样本的 TD 误差和回报。更新经验池中样本的复合优先级，为下一轮采样做准备。最终判断训练是否收敛，若收敛则输出结果，否则

重复步骤 2～7 直到训练收敛。在实际操作中，智能体与环境的交互不断进行，智能体每次根据策略采取动作并观察环境反馈，即状态的变化和奖励信号[15]。立即回报优先机制会优先采样高奖励样本，而 TD 误差优先机制则会依据当前模型的预测误差来调整样本采样的优先级。这种方式通过结合立即奖励和预测误差，优化了 DDPG 算法的采样效率和训练精度，从而提升算法的学习速度和稳定性。

2 基于优化 DDPG 算法的双 PID 控制系统

2.1 四轮机器人底盘运动学模型

研究将优化设计之后的 DDPG 算法应用于 PID 控制系统中自动调节控制参数，以此实现更精确的四轮移动机器人运动控制。机器人控制的核心目标是精确控制机器人在二维平面上的运动，包括位置、速度和方向[16-17]。四轮移动机器人与复杂的六足机器人相比，具有相对简单的动力学模型和运动学模型。运动学模型能够提供与机器人的运动行为、控制输入和状态输出之间的数学关系[18]。为此，研究建立了四轮移动机器人底盘运动学模型，即机器人的运动与轮子的运动建立的数学关系模型，如图 3 所示。

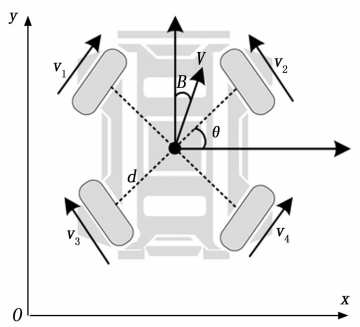


图 3 四轮移动机器人底盘运动学模型

图 3 中，假设机器人以速度 V 的角度 B 运动，系统在 x、y 方向上的移动速度分别为 v_x、v_y，如式（7）所示：

$$\begin{cases} v_x = V\sin B \\ v_y = V\cos B \end{cases} \tag{7}$$

式（7）表明机器人的整体速度，即 x 和 y 方向上的速度与轮子各自的速度之间是有一定规律。PID 控制器的任务是调节机器人在各个方向上的速度，其依据运动学模型计算出的每个轮子的速度来进行调节[19]。这一过程确保了机器人在实际运动中能精确控制运动的目标，正常情况下确保机器人朝正确的方向行驶，并且达到预定的速度。其中，4 个轮子的速度为 v_b，θ 为机器人底盘中心处测量的轮子与 x 轴的夹角，4 个轮子的速度 v_1、v_2、v_3、v_4，如式（8）所示：

$$\begin{cases} v_1 = \cos\theta \cdot v_x + \sin\theta \cdot v_y - wd \\ v_2 = \cos\theta \cdot v_y - \sin\theta \cdot v_x + wd \\ v_3 = \cos\theta \cdot v_y - \sin\theta \cdot v_x - wd \\ v_4 = \cos\theta \cdot v_x + \sin\theta \cdot v_y + wd \\ v_b = [v_1, v_2, v_3, v_4] \end{cases} \tag{8}$$

式（8）中，w 为底盘的角速度，d 为底盘中心到 4 个轮子的距离。角速度反映机器人底盘的旋转速度，而轮子的速度决定机器人的平移速度。这种旋转和移动之间的关系是调整机器人方向和速度的基础[20]。运动学模型为 PID 控制器提供了实际运动，即机器人在 x、y 方向的速度与轮子速度之间的数学关系，帮助控制器进行调节。同时，运动学模型帮助优化 PID 控制器的训练，确保控制器在调节过程中能够根据机器人的实际运动行为给出最合适的控制指令[21]。

2.2 高精度控制系统设计

在机器人底盘运动学模型基础上，研究设计了一种高精度控制器系统，以便提升机器人的运动精度和适应性。具体来说，研究通过两个独立的 PID 控制器来控制机器人的位置，两个控制器分别负责 x 轴和 y 轴的控制，分别处理机器人在这两个方向上的位置偏差，即 x'、y'。PID 控制器的设计不仅仅是两个 PID 控制器的简单堆叠，而是为 x、y 方向独立设计的控制器优化。x、y 方向可能会有不同的动态特性，且机器人在不同的方向上可能存在不同的控制需求。因此，分别使用两个独立的 PID 控制器，可以让每个方向的控制更加精细化，能够根据不同的状态来调节控制信号。该 PID 控制器由两个闭环组成：外环控制和内环控制。外环控制即 PID 控制器对机器人的位置误差进行调节，主要负责较大范围的控制任务。外环控制器首先计算出机器人当前位置与目标位置之间的误差。设目标位置为 (x_1, y_1)，当前机器人位置为 (x_2, y_2)，则 x'、y' 分别是机器人在 x、y 轴上的位置误差。外环 PID 控制器根据 x'、y' 计算控制信号，这一控制信号将被用来调整机器人的速度，使其朝向目标位置运动。外环 PID 控制器的公式，如式（9）所示：

$$\begin{cases} x' = x_1 - x_2 \\ y' = y_1 - y_2 \\ s_x = E_{p1} \cdot x' + E_{i1} \cdot \int x'\mathrm{d}t + E_{d1} \cdot \dfrac{\mathrm{d}x'}{\mathrm{d}t} \\ s_y = E_{p2} \cdot y' + E_{i2} \cdot \int y'\mathrm{d}t + E_{d2} \cdot \dfrac{\mathrm{d}y'}{\mathrm{d}t} \end{cases} \tag{9}$$

式（9）中，s_x、s_y 分别为控制器对 x、y 轴的输出信号，即机器人在这两个方向上的速度调整。E_{p1}、E_{p2} 分别为 x、y 轴方向的比例参数，E_{i1}、E_{i2} 分别为 x、y 轴方向的

积分参数，E_{d1}、E_{d2} 分别为 x、y 轴方向的微分参数。内环控制为 PID 控制器根据计算出的目标速度来调整机器人的各个轮子的速度，以确保机器人按期望速度运动。外环控制器输出的 s_x、s_y 会被传递给内环控制器，作为目标速度来调整机器人各轮子的速度。内环控制的任务是更精细地调整机器人在每个时刻的速度输出，以维持稳定的运动。内环 PID 控制器通过式（10）计算每个轮子的速度控制信号：

$$\begin{cases}\Delta s_x = E_{p1}\cdot(v_{x1}-v_{x2})+\\ E_{i1}\cdot\int(v_{x1}-v_{x2})\mathrm{d}t+E_{d1}\cdot\dfrac{\mathrm{d}(v_{x1}-v_{x2})}{\mathrm{d}t}\\ \Delta s_y = E_{p2}\cdot(v_{y1}-v_{y2})+\\ E_{i2}\cdot\int(v_{y1}-v_{y2})\mathrm{d}t+E_{d2}\cdot\dfrac{\mathrm{d}(v_{y1}-v_{y2})}{\mathrm{d}t}\end{cases}\tag{10}$$

式（10）中，v_{x1}、v_{y1} 分别为 x、y 轴方向外环控制器输出的目标速度。v_{x2}、v_{y2} 分别为 x、y 轴方向机器人当前的实际速度。由于 PID 控制器的性能取决于比例、积分和微分参数的选择。为了使机器人的控制精度和响应速度不断优化，研究通过优化算法来自动调整这些参数。结合优化算法的双 PID 控制系统设计图，如图 4 所

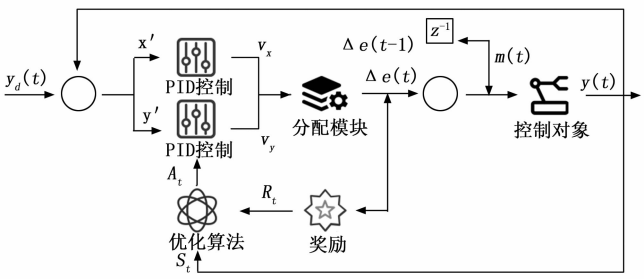


图 4　双 PID 控制系统设计图

示。图 4 中，PID 控制器的作用是根据机器人当前位置与期望位置的偏差，计算出控制信号，即轮子的速度，并根据该控制信号逐步调整机器人的位置，逼近目标位置。优化算法通过学习合适的 PID 参数，自动调整 PID 控制器的行为，进而改善控制效果。对于每个 PID 控制器，需要确定比例、积分、微分参数，设定每个控制器的参数范围为［0，10］。其中，状态空间 S_t 包括机器人的当前位置、速度、偏差等信息。动作空间 A_t 包括 PID 参数的调整范围（例如 E_{p1}、E_{p2}、E_{i1}、E_{i2}、E_{d1}、E_{d2}）。t 时刻的 A_t 关系式如式（11）所示：

$$A_t=[E_{p1},E_{i1},E_{d1},E_{p2},E_{i2},E_{d2}]\tag{11}$$

相同时刻的 S_t 关系式，如式（12）所示：

$$S_t=[v_x,v_y,x,y,V,\theta,w,B,x_e,y_e,u_{vx},u_{vy}]\tag{12}$$

式（12）中，v_x、v_y 分别为系统在 x、y 方向上的移动速度。x_e、y_e 分别在 x、y 方向上实际输出与期望输入之间的误差。u_{vx}、u_{vy} 分别在 x、y 方向上的 t 时刻输出与 $t-1$ 时刻输出之差。将机器人在 x、y 轴上的位置误差 x'、y' 作为反馈信号，结合速度误差 x_e、y_e，用来评估控制效果。奖励函数 R_t 根据误差和速度误差的大小来评估当前 PID 参数的控制效果，较小的误差或较快的响应速度得到较高的奖励。R_t 计算式，如式（13）所示：

$$\begin{cases}R_1=-0.000\ 01\\ R_2=-10,e(t)\geqslant R_1\\ R_3=0,e(t)<R_1\\ R_4=-200,e(t)>0.5\\ R_t=R_1+R_2+R_3+R_4\end{cases}\tag{13}$$

式（13）中，$e(t)$ 为系统总误差。R_1 为轮子速度的奖励，由于轮子速度的影响相对较小，因此设置为一个较小的负值。R_2 为误差低于允许区间时，给予较高的负奖励，表示误差过大时需要更强的调整。R_3 为当误差位于允许范围内时，给予中性的奖励，不做特别惩罚或奖励。R_4 为当误差超过 0.5 时，给予较大的负奖励，表明误差过大时对系统有严重影响。通过 R_t 的设计，系统鼓励减少控制误差。通过合理设计奖励函数，可以引导算法学习到更优的控制策略，从而提高机器人的表现[22]。通过独立优化的 PID 控制器，提高了控制的精细化程度，能够更好地应对不同方向上的控制需求。在训练过程中，算法通过试错法不断选择状态和动作，根据所获得的奖励反馈调整参数。每一轮训练后，优化算法会更新 PID 参数，以最小化控制误差，提高控制精度，最终达到控制精度的最优值。经过多轮训练后，优化算法将输出最优的 PID 参数，这些参数可以用来调整实际的控制系统，以获得更高的精度和更好的轨迹跟踪效果。利用从训练中获得的最佳 PID 参数，实现精确轨迹跟踪的关键就是控制轮子速度，确保机器人能够按照预定路径行驶[23]。

3　基于优化算法的机器人运动控制性能测试分析

3.1　实验平台及参数设置

实验首先准备测试设备和仿真环境，之后设定初始参数，训练优化算法，进而调整 PID 参数。之后通过设计不同参考轨迹实现跟踪测试，并在 Simulink 中进行不同控制策略的性能对比分析。研究硬件配置 Intel Core i7 CPU，NVIDIA RTX5000 显卡，内存 32 GB，存储选用固态硬盘。机器人硬件平台选择四轮机器人，通过惯性测量单元运动传感器监测机器人位置和速度[24]。软件选择 Matlab/Simulink 构建数字孪生模型，RL Toolbox 实现 DDPG 优化，采样频率 1 kHz。算法的参数演员学习率为 0.006，评论家学习率 0.001，折扣率为 0.99，软更新参数为 0.001，迭代次数为 1 000，

经验池容量为 1.0×106。

3.2 优化算法收敛性与稳定性分析

为了验证优化算法在收敛速度和稳定性方面的优势，研究将优化算法与 DDPG 算法的回报总和以及最优奖励值变化进行对比。二者的训练曲线变化和最优奖励曲线变化，如图 5 所示。

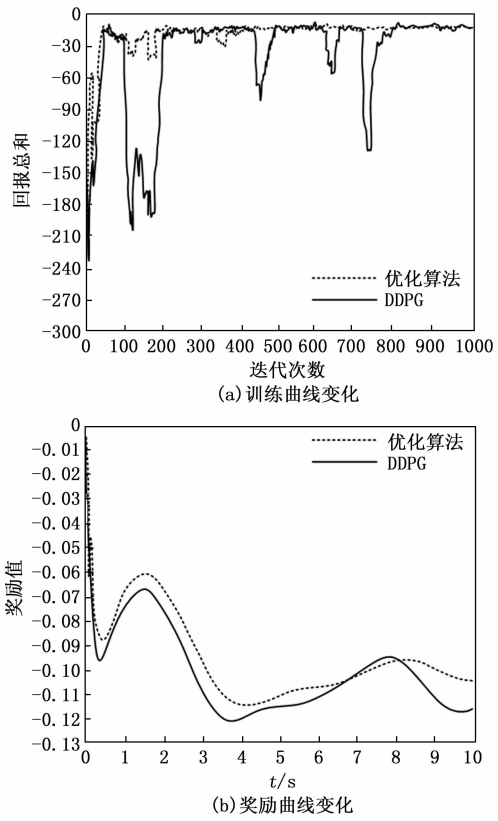


图 5 DDPG 和优化算法的训练、奖励曲线变化

图 5（a）中，在 200 次迭代时，优化算法的训练曲线趋于稳定，并在 400 次迭代时完全收敛，最终回报总和约为－15。优化算法能够快速适应环境并调整策略，从而在较短的时间内达到最佳控制策略。DDPG 算法在 1000 次迭代内表现出较大的波动，其训练过程更加不稳定，回报的累计值存在较大的起伏，直到较晚阶段才趋于稳定。DDPG 算法在调整控制策略时存在一定的不确定性，学习过程较为缓慢。优化算法的训练速度显著优于 DDPG，对比之下提升了 50%，能够在较少的迭代中找到最优策略。图 5（b）中，在前 1 s，优化算法的奖励值变化较快，且在 4 s 后奖励值变化逐渐趋于平稳，表明优化算法能够迅速找到适应环境的最优策略，并在训练过程中保持较高的奖励值。DDPG 算法在初期奖励值变化缓慢，10 s 内的奖励值大约有 79.8%的时间小于优化算法。

由于参数的选择与设置能够反映优化后的 PID 控制系统在平衡精度、响应速度和稳定性方面的有效调节。为了确定具体的 PID 参数值，研究通过优化算法在线调整两个 PID 控制器参数，相关的参数变化对比，如图 6 所示。

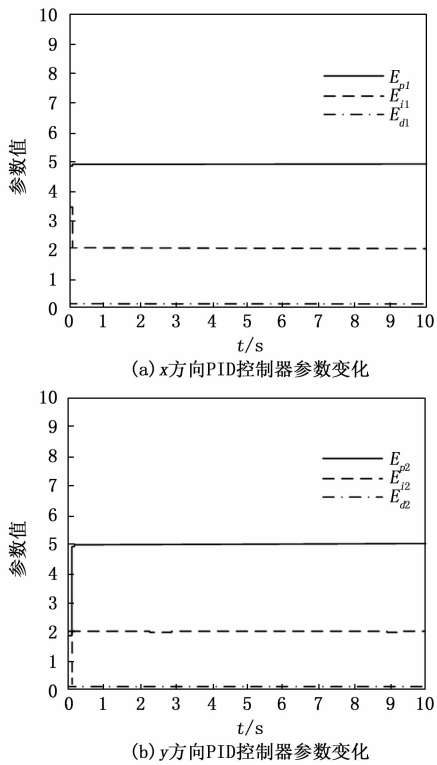


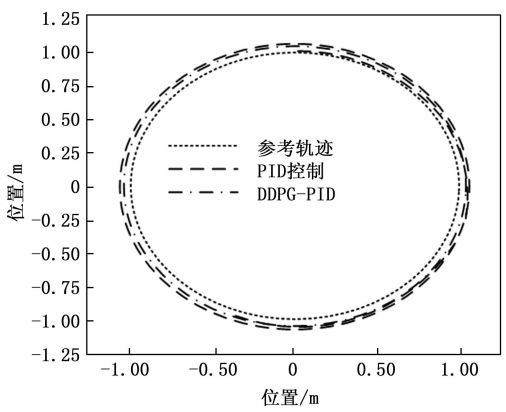
图 6 双 PID 控制系统参数变化

图 6（a）中，x 向 PID 控制器的参数 $E_{p1}=5$，$E_{i1}=2$，$E_{d1}=0.2$。此时控制器对位置误差的反应较为强烈，以快速修正误差。系统对长期误差的修正有一定程度的强度，但避免过度敏感，保持稳定性。E_{i1} 设置 2，使得机器人能够有效消除长期的微小误差，防止轨迹偏离目标。E_{d1} 较低，其主要起到平滑调整的作用，而不是主导系统的动态响应。这可以帮助避免过度反应，尤其是在误差变化较快时，确保系统的稳定性。图 6（b）中，而 y 向 PID 控制器的参数为 $E_{p2}=5$，$E_{i2}=2$，$E_{d2}=0.2$，与 x 方向上的参数设置相同。这能够保证机器人的两个方向的控制效果一致，从而达到精确、平稳的运动控制。

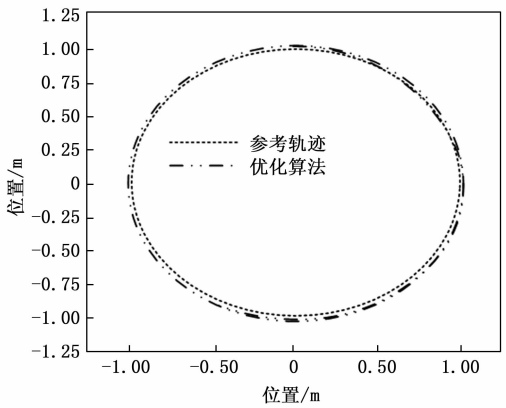
3.3 多种轨迹跟踪测试

在确定的双 PID 控制器参数下，为了验证优化算法控制的有效性，研究将其与仅用 PID 控制和 DDPG-PID 控制这两种控制算法进行比较。研究使用 Matlab 进行仿真实验，模拟机器人的圆形轨迹跟踪。研究将机器人初始位置坐标设定为（0，1），初始速度为 0，机器人从静止状态开始运动。采样时间为 1 ms，参考轨迹为单位圆，即机器人需要沿着半径为 1 的圆形轨迹进行

运动。设置为 $x=\sin t, y=\cos t$。3 种方法在跟踪轨迹上的表现，如图 7 所示。



(a) PID控制和DDPG-PID控制轨迹对比



(b) 优化算法控制轨迹

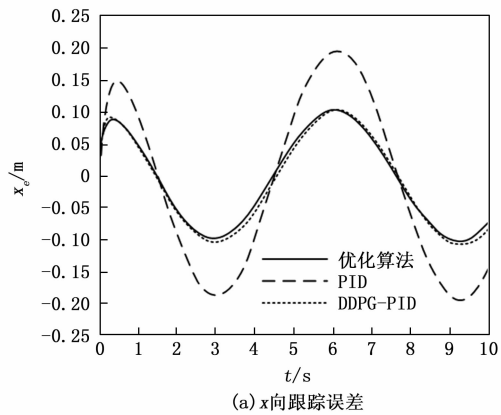
图 7　不同控制方法的跟踪轨迹

图 7（a）中，距离参考轨迹最远的为仅用 PID 控制的轨迹。原因在于使用固定的 PID 参数会在一些简单的情况下表现良好，但对于复杂的、动态变化的运动轨迹，比如机器人沿单位圆轨迹运动，PID 控制的固定参数难以适应不同状态下的需求，因此产生较大的误差。对比之下，DDPG-PID 控制通过智能算法学习和调整 PID 参数，能够较传统 PID 控制有所改进，适应性更强。但由于其仍然依赖于预设的 PID 结构，无法完全实现实时的、动态的参数调整，因此表现略逊色于优化算法控制。图 7（b）中，参考轨迹即为理想的圆形轨迹，而距离参考轨迹最近的优化算法控制下的轨迹，二者之间具有较小的误差。优化算法控制通过在线调整 PID 控制器的参数，在运行过程中实时优化，能够根据不同的误差情况和运动状态作出动态调整，表现出最佳的控制效果。这种方法的优势在于其动态适应性和误差最小化，能够有效应对复杂、动态的控制环境，从而确保机器人在各个维度上都能高精度地跟踪参考轨迹。为进一步验证算法泛化性，设计了不规则曲线、多段不同轨迹组合场景。不规则曲线采用 Lissajous 曲线生成随机轨迹[25]。多段不同轨迹组合为包含直线、圆弧、急转弯的组合轨迹[26]。两个场景下测试结果，如表 1 所示。

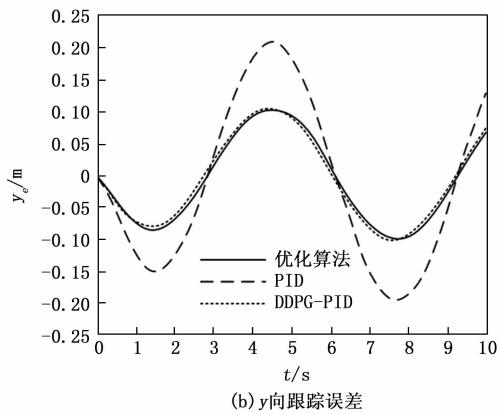
表 1　3 种优化策略性能表现结果

性能指标	控制方法	RMSE/m	最大误差/m	恢复时间/s
不规则曲线	优化算法	0.09	0.14	/
	PID	0.22	0.30	/
多段不同轨迹组合	优化算法	0.07	0.12	0.40
	PID	0.18	0.25	1.20
	DDPG-PID	0.10	0.15	0.60

表 1 中，优化算法在不规则曲线轨迹、多段不同轨迹组合中均方根误差（RMSE，root mean square error）分别为 0.09、0.10 m，显著优于 DDPG-PID、PID 方法。动态场景下优化算法的误差波动更小，具有强适应性。为验证优化算法的优势，研究对比 3 种控制方法在 x 向和 y 向跟踪误差上的表现。不同控制算法的轨迹跟踪误差对比，如图 8 所示。



(a) x向跟踪误差



(b) y向跟踪误差

图 8　误差分析

图 8（a）中，x 向上仅用 PID 控制的最大误差为 0.200 3 m。使用固定的 PID 参数控制时，机器人在 x 轴方向上会偏离参考轨迹较远，误差较大。优化算法控制的仅为 0.097 6 m，对比 DDPG-PID 的 0.104 7 m 误差降低了 9.76%。优化算法通过自动调整 PID 控制器

的参数，能够使机器人在 x 方向上更精确地跟踪参考轨迹，从而减小误差。图 8（b）中，y 向上 PID、DDPG-PID 和优化算法的跟踪误差分别为 0.209 7 m、0.108 8 m、0.108 8 m。相较 PID 控制，优化算法误差减少了大约 48.0%。综合来看，优化算法控制的效果略优于 DDPG-PID 控制，通过优化算法自适应调整 PID 参数，可以进一步提高跟踪精度，减少误差。在不同控制条件下，优化算法能展现更好的性能。

3.4 优化策略性能分析

研究对设计的优化算法控制系统进行性能测试，分别采用文献［27］、文献［28］的 DDPG 优化策略，在四轮机器人运动控制中进行性能测试。文献［27］为引入自注意力机制、后见经验算法优化 DDPG，即策略 1。文献［28］为自适应奖励函数结合 DDPG，即策略 2。通过对稳定性、响应速度、平均奖励值、误差等指标的综合评估，展示不同优化策略在机器人运动控制中的表现差异。其中误差包括稳态误差和动态误差，表示系统输出与目标值的差异。3 种优化策略下 DDPG 算法在四轮机器人运动控制中的性能表现，如表 2 所示。

表 2 3 种优化策略性能表现结果

性能指标	优化算法	策略 1	策略 1
稳定性	高，波动小	中等，受参数影响较大	低，奖励值波动较大
响应速度	快，稳定响应	中等，响应较慢	较慢，初期波动较大
平均奖励值	45	35	30
稳态误差	0.02	0.08	0.10
动态误差	0.05	0.15	0.20
控制精度	高，误差小	中等，依赖于 PID 参数调节	低，初期精度不稳定

表 2 中，机器人实际运动控制效果能够从稳定性、响应速度、误差、控制精度等方面体现。优化算法控制系统表现出较高的稳定性，波动较小，并且响应速度较快，能够在短时间内对环境变化做出有效反应，维持稳定的响应。策略 1 响应速度适中，后见经验算法可以提高稀疏奖励情况下的探索效率。但由于其计算复杂度高，导致响应速度稍慢。策略 2 平均奖励值仅为 30，因自适应奖励函数的调整复杂度较高，导致策略学习的不稳定性，从而降低了最终的平均奖励。在稳态误差和动态误差两个指标上，优化算法控制系统表现最佳，稳态误差为 0.02，动态误差为 0.05，显示出较高的控制精度。相比之下，策略 1、策略 2 控制系统的误差较大。策略 1 的算法稳态误差为 0.08，由于后见经验算法能够利用成功和失败的经验进行学习，但其策略稳定性仍存在一定波动，影响稳态误差。尽管策略 1 的自注意力机制有助于提高感知能力，但机器人在初始探索阶段仍存在较大的误差波动，动态误差达到 0.15。策略 2 稳态误差和动态误差分别为 0.10、0.20，因其自适应奖励函数无法快速适应所有场景，导致稳态误差较大。同时该系统面临动态干扰，策略调整需要更多时间，导致误差波动较大。

4 结束语

为了进一步提升四轮机器人的运动控制精度，研究采用双机制优化 DDPG 算法，并将其应用到双 PID 控制系统中。结果表明，优化算法控制的两个 PID 控制器的参数设置为 E_{p1}、E_{p2} 为 5，E_{i1}、E_{i2} 为 2，E_{d1}、E_{d2} 为 0.2。优化算法在控制过程中能够根据环境和误差的变化实时调整 PID 控制器的参数，从而减少控制过程中产生的误差，并提高轨迹跟踪的精度。优化算法在动态环境中展现出较强的适应能力，在复杂的机器人控制任务中能够更好地应对误差积累问题，从而有效提高机器人的控制精度。与传统 PID 控制和基于 DDPG 的 PID 控制方法相比，优化算法的优势在于其能够在控制过程中动态调整参数，保证了系统的精确性和稳定性。传统 PID 控制由于固定的控制器参数，常常面临在不同环境和任务下性能不稳定的问题。而 DDPG-PID 控制虽然能够通过强化学习方法优化控制器的行为，但在面对环境变化较大时，依然存在一定的误差波动。而优化算法通过结合双机制的优化方法，能够实时反馈并调整控制参数，因此在实际应用中表现出更高的精度和稳定性。优化算法相比于单纯的 DDPG 算法，其训练速度提高了 50%。这一改进使得优化算法在短时间内能够快速收敛，并在整个训练过程中展现出较小的波动性，收敛过程更加平稳。根据实验结果，在相同的训练时间内，优化算法的奖励值有 79.8%的时间高于 DDPG 算法，显示出其更高的训练效率。这使得优化算法特别适合用于那些需要快速实现控制策略的机器人运动任务，在需要提高响应速度和跟踪精度的复杂场景中具有明显的应用优势。优化算法有效提高了控制精度，并增强了系统的稳定性和可靠性。但研究没有考虑山地、盆地等不同地形机器人的运动控制性能的影响，未来可以引入地形感知模块与自适应控制策略，进一步优化机器人在复杂环境中的表现。

参考文献：

［1］ BRUZZONE L, NODEHI S E, FANGHELLA P. WheTLHLoc 4W: Small - scale inspection ground mobile robot with two tracks, two rotating legs, and four wheels［J］. Journal of Field Robotics, 2024, 41 (4):

1146 - 1166.
[2] CHOTIKUNNAN P, CHOTIKUNNAN R, NIRAPAI A, et al. Optimizing membership function tuning for fuzzy control of robotic manipulators using PID-Driven data techniques [J]. Journal of Robotics and Control (JRC), 2023, 4 (2): 128 - 140.
[3] BHOURJI R S, MOZAFFARI S, ALIREZAEE S. Reinforcement learning ddpg - ppo agent-based control system for rotary inverted pendulum [J]. Arabian Journal for Science and Engineering, 2024, 49 (2): 1683 - 1696.
[4] CAO H, LI Y, LIU C, et al. ESO-based robust and high-precision tracking control for aerial manipulation [J]. IEEE Transactions on Automation Science and Engineering, 2023, 21 (2): 2139 - 2155.
[5] PENG Z, LIU E, PAN C, et al. Model-based deep reinforcement learning for data-driven motion control of an under-actuated unmanned surface vehicle: Path following and trajectory tracking [J]. Journal of the Franklin Institute, 2023, 360 (6): 4399 - 4426.
[6] GUO Y, LI B, SPANOGIANNOPOULOS S, et al. DDPG-based controlling algorithm for upper limb prosthetic shoulder joint [J]. International Journal of Control, 2024, 97 (5): 1083 - 1093.
[7] 李　静，罗　征，闫振平，等. 基于改进机器学习的图书馆机器人自主避障控制研究 [J]. 计算机测量与控制，2024，32 (9)：200 - 205.
[8] DESHPANDE A M, HURD E, MINAI A A, et al. Deep-cpg policies for robot locomotion [J]. IEEE Transactions on Cognitive and Developmental Systems, 2023, 15 (4): 2108 - 2121.
[9] 刘　静，杨　雪，陈　伟，等. 无缆自治水下机器人运动控制参数整定方法 [J]. 计算机仿真，2023，40 (3)：421 - 425.
[10] XIONG Y, LIU S, ZHANG J, et al. Nonlinear control strategies for 3-DOF control moment gyroscope using deep reinforcement learning [J]. Neural Computing and Applications, 2024, 36 (12): 6441 - 6465.
[11] QIU C, WU Z, WANG J, et al. Multiagent-Reinforcement-Learning-Based stable path tracking control for a bionic robotic fish with reaction wheel [J]. IEEE Transactions on Industrial Electronics, 2023, 70 (12): 12670 - 12679.
[12] 何　佳. 环境信息实时感知的图书馆搬运机器人自动化控制系统 [J]. 计算机测量与控制，2024，32 (7)：181 - 188.
[13] 侯茂林，马春燕，庞　健，等. 小型水下机器人运动控制系统设计 [J]. 现代电子技术，2023，46 (22)：53 - 57.
[14] XIN G, MISTRY M. Optimization-based dynamic motion planning and control for quadruped robots [J]. Nonlinear Dynamics, 2024, 112 (9): 7043 - 7056.
[15] 李艳红. 基于人工智能技术的机器人运动控制系统设计 [J]. 现代电子技术，2024，47 (10)：117 - 122.
[16] HAN L, CHEN X, YU Z, et al. A heuristic gait template planning and dynamic motion control for biped robots [J]. Robotica, 2023, 41 (2): 789 - 805.
[17] 陈卓凡，周　坤，秦菲菲，等. 基于改进量子粒子群优化算法的机器人逆运动学求解 [J]. 中国机械工程，2024，35 (2)：293 - 304.
[18] MAKIN T R, MICERA S, MILLER L E. Neurocognitive and motor-control challenges for the realization of bionic augmentation [J]. Nature biomedical engineering, 2023, 7 (4): 344 - 348.
[19] 李志文，程志江，杜一鸣，等. 基于 ROS 的清洁机器人运动控制研究 [J]. 计算机仿真，2023，40 (4)：455 - 460.
[20] RAWAT D, GUPTA M K, SHARMA A. Intelligent control of robotic manipulators: a comprehensive review [J]. Spatial Information Research, 2023, 31 (3): 345 - 357.
[21] SON J, KANG H, KANG S H. A review on robust control of robot manipulators for future manufacturing [J]. International Journal of Precision Engineering and Manufacturing, 2023, 24 (6): 1083 - 1102.
[22] 刘　行，黄庭安，董云龙，等. 基于示教融合的深度强化学习机器人化齿轮装配算法 [J]. 控制工程，2023，30 (7)：1308 - 1316.
[23] SCHUCHERT P, KARIMI A. Frequency-domain data-driven position-dependent controller synthesis for cartesian robots [J]. IEEE Transactions on Control Systems Technology, 2023, 31 (4): 1855 - 1866.
[24] 李　肖，李世其，韩　可，等. 面向实时自避碰的双臂机器人力矩控制策略 [J]. 信息与控制，2023，52 (2)：211 - 234.
[25] 王　彦，蒋　超，朱　伟，等. 基于 Lissajous 输出拟合的非平衡 M-Z 干涉仪相位解调 [J]. 仪器仪表学报，2023，44 (2)：84 - 92.
[26] 张　鑫，秦东晨，谢远龙，等. 基于双环自适应滑模的移动机器人轨迹跟踪控制 [J]. 合肥工业大学学报（自然科学版），2024，47 (1)：13 - 20.
[27] 王凤英，陈　莹，袁　帅，等. 自注意力机制结合 DDPG 的机器人路径规划研究 [J]. 计算机工程与应用，2024，60 (19)：158 - 166.
[28] 崔　立，宋　玉，张　进. 基于自适应 DDPG 方法的复杂场景下 AUV 动动对接 [J]. 船舶工程，2023，45 (8)：8 - 14.