Chapter 4
Trajectory Optimization Control System of Intelligent Robot Based on Improved Particle Swarm Optimization Algorithm

Ziyu Hu

Abstract Trajectory optimization is a hot topic in the field of intelligent robots, whose task is to plan the optimal motion trajectory that passes through a specified point and satisfies constraints such as velocity and acceleration based on a given target trajectory point. In the time optimization problem of robots, particle swarm optimization (PSO) has been widely applied in the time optimization problem of robotic arms due to its simple structure and adjustable parameters. This article conducts research on the intelligent robot trajectory optimization control system based on PSO, and the results show that through experimental data of three joints, it can be found that the optimized trajectory motion time has been reduced by an average of about 50%, achieving the expected goal. Compared with before optimization, PSO has stronger global search ability, faster convergence speed, and good stability, which can effectively improve robot work efficiency and maintain smooth operation. By dynamically adjusting the value of the learning factor in the particle swarm optimization algorithm through PSO, the particle swarm can search for the optimal value in a short period of time in the early stage of iteration and can quickly and accurately converge to the optimal solution in the later stage of iteration.

4.1 Introduction

The application engineering research of intelligent robots has attracted worldwide attention. Outdoor small intelligent mobile robots are a kind of engineering service robots, which have broad application prospects and can be applied to environmental cleaning, agricultural and forestry plant protection, resource exploration and other occasions. The motion control of intelligent robots can be divided into three categories: point stabilization, path tracking and trajectory tracking. In trajectory tracking control, the tracking path is related to time, so it is the most complicated. At present,

Z. Hu (✉)
School of Economics and Management, Shanghai Institute of Technology, Shanghai 200235, China
e-mail: 1292107005@qq.com

R. Kountchev et al. (eds.), *Proceedings of International Conference on Artificial Intelligence and Communication Technologies (ICAICT 2023)*, Smart Innovation, Systems and Technologies 368, https://doi.org/10.1007/978-981-99-6641-7_4

there are mainly control based on kinematic model, control based on dynamic model and intelligent control [1]. At present, there are two kinds of trajectory planning: one is to optimize the time, and choosing the motion time as the optimization object can improve the working efficiency of the robot; the other is to optimize the system energy and choose the system energy as the optimization object, which can reduce the energy loss of the robot and prolong the service life [2]. Trajectory optimization is a hot issue in the robot field. Its task is to plan the optimal trajectory that passes through the specified point and meets the constraints of speed and acceleration according to the given target trajectory point. In the robot time optimization problem, PSO is widely used for its simple structure and adjustable parameters [3]. In this paper, the trajectory optimization control system of intelligent robot is studied based on PSO, and the time is optimized by PSO to find out the optimal trajectory. Experiments show that when selecting the optimal position of particle swarm, the poor solution is accepted with a certain jump probability to jump out of the local extreme value, and after a certain number of iterations, if the optimal individual in the swarm is found to be not obviously optimized, particles with different concentrations are inhibited or promoted [4, 5]. Fully considering the dynamic characteristics of the mobile robot, it “retreats” to the whole system step by step. Based on the dynamic model of the mobile robot, a globally stable trajectory tracking control law is designed, which has engineering application value. By dynamically adjusting the value of learning factor in PSO, the particle swarm can search for the optimal value in a short time at the beginning of iteration and converge to the optimal solution quickly and accurately at the end of iteration [6].

4.2 Research on Trajectory Planning of Intelligent Robots Based on Improved Particle Swarm Optimization Algorithm

4.2.1 Trajectory Optimization Control System

Intelligent robots have various internal and external information sensors, such as vision, hearing, touch and smell. As shown in Fig. 4.1, intelligent robots not only have receptors, but also effectors as a means of acting on the surrounding environment. This is the muscle, also known as the stepper motor, which moves the hands, feet, long nose, antennae and so on. From this, it can also be seen that intelligent robots must have at least three elements: sensory elements, reaction elements and thinking elements.

The difference between intelligent robots and industrial robots is that they have the ability to perceive, recognize, reason and judge like humans do. You can modify the program within a certain range based on changes in external conditions, which means it can adapt to changes in external conditions and make corresponding adjustments to itself. However, the principles for modifying the program are predetermined by

Fig. 4.1 Intelligent robot

individuals. This type of primary intelligent robot has already possessed a certain level of intelligence. Although it does not yet have the ability to automatically plan, it has also begun to mature and reach a practical level. The trajectory planning is divided into two stages during implementation. The first stage is called path planning, which refers to the various paths that require the robot to move to a designated position. How to move to the designated position, such as through a straight line or arc to move to the designated position. The second stage is called trajectory tracking, which means that for a known path or trajectory, the robot cannot perform motion uniformly according to it, but uses approximation to complete the given trajectory [7]. However, trajectory planning has always been a research hotspot in the field of intelligent robot control [8]. In this regard, this article will conduct research on the trajectory optimization control of intelligent robots based on PSO. The basic PSO adopts a "speed position" search model to solve optimization problems.

Elementary particle swarm optimization algorithm has fast convergence performance in the initial stage. The performance of this controller depends on the selection of controller parameters, and it takes a lot of work to determine these parameters by repeated experiments [9]. This article designs an intelligent robot trajectory optimization control system based on PSO and uses an improved particle swarm optimization algorithm for parameter optimization calculation. The overall structure of the controller is shown in Fig. 4.2.

In the trajectory optimization control system of the improved particle swarm optimization algorithm, the motion time of robot joints is regarded as particles in the search space, and each particle has a position attribute and a speed attribute, and the

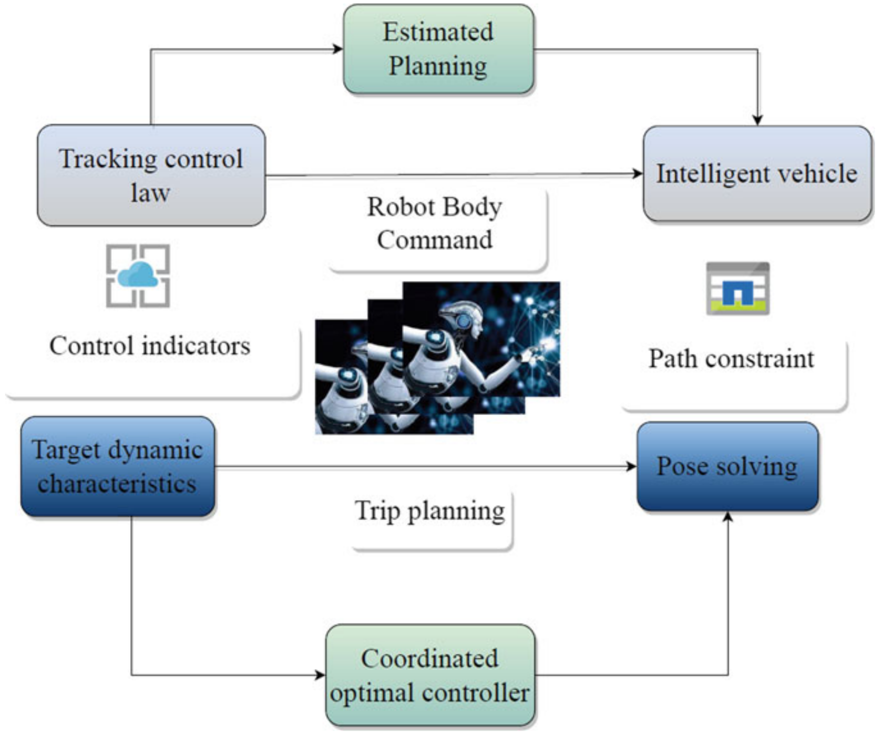


Fig. 4.2 Trajectory optimization control system

optimal time is sought by the fitness function [10]. The particle swarm adjusts the next evolutionary direction according to the individual optimal position and the global optimal position, but when it is local optimal, all particles will be influenced by it and move toward the local optimal position, which will lead to the rapid convergence of the particle swarm, resulting in local extremum or stagnation.

4.2.2 Optimization Goal Establishment

The entire particle swarm is a large whole, treating each particle as a separate small individual. Each particle randomly finds a location that may be the best within the feasible domain. The target point is determined as the global optimal solution, and the motion process of the particle swarm can be seen as the iterative evolution process of the algorithm. In particle swarm optimization, each particle exists as an independent individual, representing an optimization mechanism that seeks the optimal solution through mutual cooperation between particles. The PSO algorithm is an intelligent optimization algorithm that is very simple, easy to understand and does not require complex calculations or excessive parameter adjustments. Its robustness is stable. Particle swarm optimization (PSO) algorithm is prone to falling into the defects of extreme values and declining particle swarm diversity during the iteration process for improvement, mainly through parameter improvement, adjusting particle states and integrating with other algorithms. Scholars often make improvements based on inertia weight coefficients.

When the intelligent robot is working, whether it meets the requirements is mainly measured by running time, energy consumption and pulsating impact. According to different iterative stages, the learning factor is dynamically adjusted to prevent the particle swarm from falling into local optimum in the initial rapid aggregation stage, and to ensure that the global optimal solution can be quickly found in the later search stage while searching in a large range [11]. Aiming at the optimization of intelligent robot's time pulse impact, the time-based objective function S_1 and the pulse impact-based objective S_2 are constructed:

$$\min S = T = \sum_{i=0}^{n-1} (t_{i+1} - t) \tag{4.1}$$

$$S_2 = \sum_{n=1}^{N} \sqrt{J_i^2 \mathrm{d}t} \tag{4.2}$$

Among them: T stands for the total running time of intelligent robot, which can be used to measure the running efficiency; the J represents the pulsating impact and measures the smoothness of the robot trajectory.

The objective function is normalized by setting the weight coefficient, and the fitness function is defined as:

$$f = \alpha_1 S_1 + \alpha_2 \beta S_2 \tag{4.3}$$

where α_1 and α_2 are weight coefficients and satisfy $\alpha_1 + \alpha_2 = 1$; β is an elastic adjustment factor, and β exists to balance the difference in order of magnitude between pulsating impact and action time.

In order to perform time optimal trajectory planning for articulated industrial robots, it is necessary to optimize particle parameters and find the optimal solutions for each particle, in order to obtain the optimal solution for the entire optimization process. The optimal value of the particle swarm has a significant impact on the behavior of the entire particle swarm. The idea of simulated annealing is used to select the optimal solution of the particle swarm [12, 13]. In the early stages of optimization, it is acceptable to accept inferior particles as particle swarm particles, maintaining particle diversity. In the later stage of optimization, only high-quality particles are accepted as the optimal particles, so that the optimization converges to the optimal value.

4.2.3 Particle Velocity and Position Updates

PSO is an algorithm based on traditional PSO, but the number of objects to be solved is different. After the target particle swarm is initialized, the particle characteristics in the particle swarm are represented by position, speed and fitness. If the robot needs to accurately track a given trajectory, then the controller must control the manipulator's executive end to move along this trajectory. However, because the robot is highly nonlinear and has complex constraints, Coriolis force and centrifugal force have great influence on the movement of the whole system at high speed. It is generally impossible for a robot to accurately track a given trajectory. So most of the trajectory tracking is realized by approximation. In order to ensure that the robot's end gripper can reach Cartesian space coordinates, the robot is solved by forward kinematics and the space coordinate position of its starting point and the space coordinate position of its ending point are obtained. Compared with the different optimal individual fitness produced by each iterative update, the global optimal solution G_{best} is determined, and the iterative expressions of the velocity and position of particle i are as follows:

$$\upsilon_i^{k+1} = \varphi_1 R_1\left(P - X_i^k\right) + \varphi_2 R_2\left(G - X_i^k\right) \tag{4.4}$$

Among them, υ_i^k represents the velocity of the i particle at the k iteration; X_i^k is the position after the k iteration. R_1 and R_2 are random numbers between (0, 1). P_1 and P_2 represent the optimal positions for particle fitness.

The selection of particle swarm fitness function is the key to the optimization results. Due to the existence of many different evaluation systems and performance indicators in the control process, such as stability, controllability, convergence speed, steady-state characteristics and dynamic characteristics. For robots with joints, corresponding trajectories are set in joint space. If more points are selected, the distance between adjacent two points will be smaller and the accuracy will be higher, but this also leads to the need for more inverse operations. The lines and arcs implemented in this section are implemented in this way. Therefore, different emphasis points in the optimization process will be reflected in the different characteristics presented on the tracking trajectory.

4.3 Simulation Experiment and Result Analysis

4.3.1 Simulation Conditions and Parameter Settings

In this paper, the improved particle swarm optimization algorithm, the standard particle swarm optimization algorithm, the inertia weight PSO algorithm and the compressibility factor PSO algorithm are used to carry out path planning simulation experiments through MATLAB. Due to multiple robots performing path planning tasks in the same operating environment, there may be resource grabbing and collaborative cooperation between robots. Therefore, we need to allocate tasks reasonably, as well as prioritize node interaction, evaluation and decision-making.

During the experiment, we need to consider not only the conditions set in the particle swarm optimization algorithm, but also the normal conditions of industrial robots, which should not exceed the normal range of robots. PSO is an algorithm to ensure global search. In order to further improve the global search ability of the algorithm, improve the search performance for solving multimodal optimization problems and improve the ability of searching in a large range in the later stage of the algorithm. In this paper, the operation strategy of randomly selecting the best individual in the group is put forward to avoid the diversity of the group being too small, so as to improve the global search ability of the algorithm.

In each iteration, the particle itself will constantly adjust by tracking two extreme values. In the optimization process, firstly, the global optimal particle is obtained from twenty groups of particles by comparing fitness values. The particles are continuously updated through n iterations, and the optimal solution particle with constraints is calculated, which is the optimal time particle. So the optimal solution found from these particle neighbors is called local extremum. In the process of particle optimization, it is impossible for particles to always get a better solution only by tracking the global extremum or local extremum. The problem is solved satisfactorily when particles follow the global extreme value or local extreme value while following the individual extreme value.

4.3.2 Simulation Result

Starting from the active motion of the robot, an improved particle swarm optimization is used to solve the optimal detection point of the robot, so as to make the location and tracking of the trajectory target of the intelligent robot more accurate. In theory, the bottleneck of inaccurate distance measurement in existing positioning has been overcome, effectively solving the problem of inaccurate distance measurement in trajectory positioning accuracy. In this paper, the most appropriate weight coefficient will be selected according to the actual engineering needs to minimize the fitness function, so as to achieve the optimal solution of both. If collision needs to be avoided in the motion space of the robot, the best way is to first plan a trajectory that avoids obstacles. This trajectory is usually preplanned and stored in memory through teaching or calculation, and only needs to track the motion of this path during operation. But when the robot reaches its working position, the path of obstacles to avoid becomes short and difficult to control. This makes it very difficult to obtain these trajectories through teaching or calculation, especially in situations where the execution time is very short.

To demonstrate the stability of PSO, taking Joint 3 as an example, multiple optimizations were conducted before and after optimization. The optimization time convergence diagram based on PSO is shown in Figs. 4.3, 4.4 and 4.5. In the early stage of the experimental process, the particle swarm can search for the optimal value in a short time, while also ensuring fast and accurate convergence to the optimal solution in the later stage.

Analyzing the data in Figs. 4.3, 4.4 and 4.5, it can be seen that there is a significant difference between before and after optimization. In the time convergence results of Joint 1, it takes 7 s to complete before optimization, and only about 4 s to complete after optimization. In the time convergence results of Joint 2, it takes 9 s to complete before optimization, and only about 5 s to complete after optimization. In the time

Fig. 4.3 Time convergence graph of Joint 1

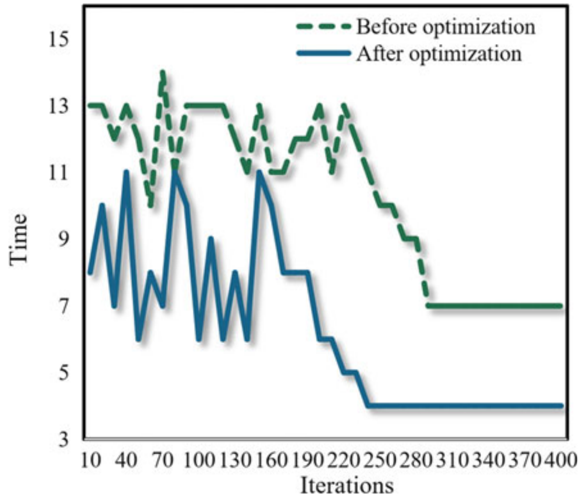
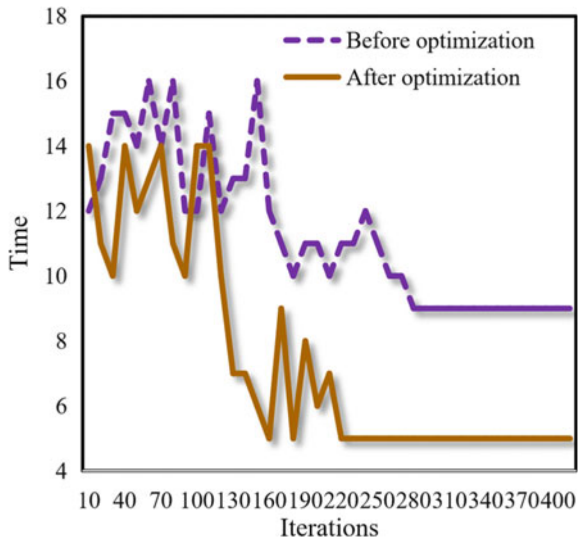


Fig. 4.4 Time convergence graph of Joint 2



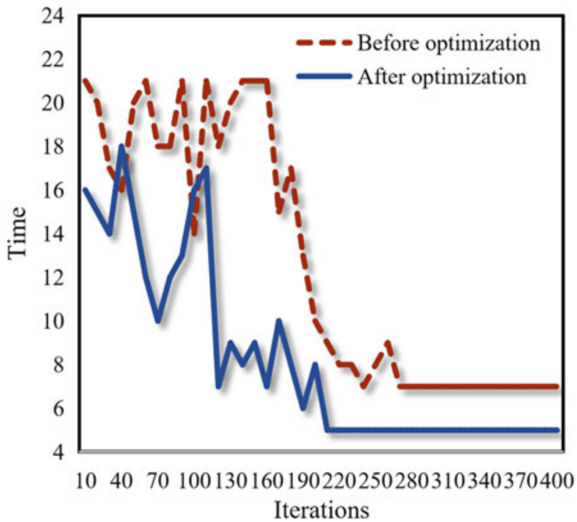


Fig. 4.5 Time convergence diagram of Joint 3

convergence results of Joint 3, it takes 7 s to complete before optimization, and only about 5 s to complete after optimization. Through the experimental data of the above three joints, it can be found that the optimized trajectory motion time has been reduced by an average of about 50%, achieving the expected goal.

4.4 Conclusions

In this paper, the trajectory of intelligent robot constructed by PSO is proposed. Combined with polynomial programming method, the optimal design of robot's time and pulse impact is carried out by improving hybrid PSO, which effectively shortens the movement time and has good stationarity, and effectively improves the robot's working performance. When the intelligent robot is working, whether it meets the requirements is mainly measured by running time, energy consumption and pulsating impact. According to different iterative stages, the learning factor is dynamically adjusted, so that the particle swarm can avoid falling into local optimum in the initial rapid aggregation stage, and the global optimal solution can be quickly found in the later search stage while searching in a large range. When designing the path, the execution end of the robot is often regarded as a point. But in fact, the size of the robot's hand is relatively large, and it moves in space with a certain volume during its movement, not a line. To avoid collision, it is necessary to ensure that the volume corresponding to this space trajectory does not intersect with obstacles. The results show that through the experimental data of three joints, it can be found that the optimized trajectory motion time is shortened by about 50% on average, which achieves the expected purpose. The results show that compared with before optimization, PSO has stronger global search ability, faster convergence speed and good stability, which can effectively improve the working efficiency of the robot and maintain the running stability.

Acknowledgements National Key R&D Program Funded Project "Technology for Trading Property Rights of Scientific and Technological Achievements and Data Resources" (2017YFB1401100).

References

1. Zhou, L., Chen, K., Dong, H., Chi, S.K., Chen, Z.: An improved beetle swarm optimization algorithm for the intelligent navigation control of autonomous sailing robots. IEEE Access **9**, 5296–5311 (2020)
2. Yan, M., Yuan, H., Xu, J., Yu, Y., Jin, L.B.: Task allocation and route planning of multiple UAVs in a marine environment based on an improved particle swarm optimization algorithm. EURASIP J. Adv. Signal Process. **2021**, 94 (2021)
3. Ma, T., Liu, S., Xiao, H.: Location of natural gas leakage sources on offshore platform by a multi-robot system using particle swarm optimization algorithm. J. Nat. Gas Sci. Eng. **84**, 103636 (2020)
4. Xu, L., Song, B., Cao, M.: An improved particle swarm optimization algorithm with adaptive weighted delay velocity. Syst. Sci. Control Eng. **9**(1), 188–197 (2021)
5. Lei, J.H., Wan, B., Liu, M., et al.: The design of four-rotor human-simulated intelligent controller based on particle swarm optimization algorithm. Comput. Simul. **46**(14), 16–23 (2022)
6. Gao, Q.: Intelligent vehicle global path planning based on improved particle swarm optimization. Open Access Libr. J. **5**, e4587 (2018)
7. Mahmoodabadi, M.J., Taherkhorsandi, M.: Intelligent control of biped robots: optimal fuzzy tracking control via multi-objective particle swarm optimization and genetic algorithms. AUT J. Mech. Eng. **4**(2), 183–192 (2020)
8. Tao, Q., Sang, H., Guo, H., Wang, P.: Improved particle swarm optimization algorithm for AGV path planning. IEEE Access **9**, 33522–33531 (2021)
9. Wu, Q., Zhu, Z., Yan, X., Gong, W.Y.: An improved particle swarm optimization algorithm for AVO elastic parameter inversion problem. Concurr. Comput. Pract. Exp. **31**(9), e4987 (2019)
10. Selma, B., Chouraqui, S., Abouassa, H.: Optimal trajectory tracking control of unmanned aerial vehicle using ANFIS-IPSO system. Int. J. Inf. Technol. **12**, 383–395 (2020)
11. Guo, Y., Fang, X., Dong, Z., Mi, H.L.: Research on multi-sensor information fusion and intelligent optimization algorithm and related topics of mobile robots. EURASIP J. Adv. Signal Process. **2021**, 111 (2021)
12. Yu, Y., Xu, Y., Wang, F., Li, W.S., Mai, X.M., Wu, H.: Adsorption control of a pipeline robot based on improved PSO algorithm. Complex Intell. Syst. **7**, 1797–1803 (2021)
13. Wang, M.: Real-time path optimization of mobile robots based on improved genetic algorithm. Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng. **235**(5), 646–651 (2021)