Research Article

Improved Genetic Algorithm for Intelligent Grinding Trajectory of Industrial Robot Sensor

Yaping Lu

Department of Mechanical and Industrial Robotics, Applied Technology College of Soochow University, Suzhou, Jiangsu 215325, China

Correspondence should be addressed to Yaping Lu; 202026619@stu.ncwu.edu.cn

Received 15 December 2021; Revised 31 January 2022; Accepted 10 February 2022; Published 26 March 2022

Academic Editor: Pradeep Kumar Singh

Copyright © 2022 Yaping Lu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to study the improved genetic algorithm for intelligent grinding trajectory of industrial robot, a trajectory planning method with optimal energy consumption is proposed. Firstly, the trajectory of the robot is regarded as a series of type value points in space, and each adjacent type value point is connected by a quintic $B$-spline curve to obtain the trajectory function of the robot. Then, the kinetic energy is taken as the target energy consumption function, and the kinematic and dynamic constraints of each joint are considered at the same time. Finally, the genetic algorithm is improved to optimize the objective energy consumption function. The improved genetic algorithm improves the operation efficiency, local search ability, and real-time performance of the algorithm. Front and rear times during handling were 15.034 s and 17.456 s for 2.422 s. The optimal secondary trajectory planning algorithm is called in the single workstation of the welding robot and the robot welding the bucket with spatial straight line and spatial curve, respectively. Through the time difference, the time used in the spatial linear welding is 5.462 s, and the spatial curve welding time is 12.981 s. The proposed algorithm can be applied to the production of various industrial sensor robots such as welding robot, cutting robot, and spraying robot and improves the working efficiency of robot operation. In addition, the genetic method of multichromosome structure also has a certain reference for solving the general GSTP problem.

1. Introduction

With the development of robot technology, the research and application of industrial sensor robot are more and more extensive, which has changed the face of traditional industrial production. Early industrial robots mostly adopted the form of “teaching reproduction,” which had defects such as high production cost, low production efficiency, and damage to the health of operators caused by harmful working environment, which led to the research on off-line programming technology of industrial robots at home and abroad [1]. Nowadays, industrial robots are widely used in all walks of life. However, they need to consume a lot of electric energy in the working process. How to effectively reduce their energy consumption without reducing their work efficiency becomes very important (Figure 1). Optimizing the movement trajectory of industrial robots can reduce their energy consumed well. One of the industrial robot control problems is optimal grinding trajectory planning, which aims to plan a smooth optimal motion trajectory that interpolates points through a given path and satisfying the boundary constraints. Optimizing the motion trajectory of industrial robots can well reduce their energy consumption [2] The research on the trajectory planning algorithm of the robot can effectively promote the work efficiency of the robot. Therefore, the international scholars who study the subalgorithm keep introducing new algorithms. The trajectory planning of the robot can be carried out in joint space or Cartesian space. If it is carried out in joint space, the joint space conversion of each working path point must be carried out first. These path points can be transformed into the angle value of each joint by inverse kinematic operation. After these working path points are transformed, they form each group of path nodes. One of the optimal control problems of industrial robots is optimal trajectory planning. Its purpose is to plan a smooth optimal trajectory through a
given path interpolation point and meet the boundary constraints. The use of industrial robots can not only increase the output and texture of products. At the same time, it can also protect personal safety, improve the working environment, save raw materials, reduce work intensity, and improve labor productivity. Therefore, it is of special significance. It is the development requirement of large-scale production and the basic unit of factory automation and flexible manufacturing system [3]. It integrates many elements such as control, communication, machinery, design, and software. It is a combination of computer science, software engineering, mechanical engineering, control engineering, and other technologies and sciences. In industrial automation or in extremely dangerous situations, it can replace people with boring and repeated long-term work such as welding, coating, stamping, machinery, and devices. Future robots should have high-level capabilities such as perception, planning, and control. Therefore, robot trajectory planning is an important part of the planning module, which has important practical significance in improving the execution efficiency of industrial robots, reducing the errors of industrial robots in actual operation, and making the robot trajectory stable, continuous, and smooth [4].

2. Literature Review

Dong, Yang, and others proposed time optimal robot trajectory planning. However, under the same trajectory, the robot action time is minimized, but its motion speed is increased, which may not effectively reduce energy consumption [5]. Dian et al. proposed the trajectory planning with optimal energy consumption, calculated the motion trajectory through spline function, and established the energy consumption model [6]. Batista et al. introduced the robot dynamic planning algorithm. However, the trajectory acceleration and joint torque generated by the above algorithm are not continuous, which will lead to the decline of trajectory accuracy, and the robot may vibrate and damage. At the same time, the controller cannot correct the path error caused by interference or modeling [7]. Therefore, in order to obtain an effective solution, Mohammad and others have made certain restrictions on the acceleration. Although the acceleration is not strictly optimal in time, the robot can effectively execute the expected trajectory [8]. In order to make the robot generate continuous trajectory, smooth trajectory is mostly used in the literature of trajectory planning. Ding et al. interpolated with B-spline curve [9]. Ahmad et al. introduced the optimal time trajectory planning algorithm and connected the important nodes in the joint space of industrial robot with high-order polynomial curve to obtain a satisfactory working trajectory [10]. Sun et al. obtained the joint variables in the joint space according to the inverse kinematics from the pose of the robot end effector and then interpolated according to the expected curve trajectory [11].

In this paper, the trajectory is obtained by B-spline function, the kinetic energy is taken as the target energy consumption, and the optimization results are obtained according to the
constraints. In the process of optimization, the improved genetic algorithm is used to overcome its shortcomings such as poor local search ability and premature phenomenon, so that the final optimization result is reliable and practical.

3. Objective Function Optimization Based on the Improved Genetic Algorithm

The genetic algorithm is an evolutionary algorithm that imitates the selection and genetic mechanism of nature. It can overcome the disadvantage of being easy to fall into the trap of local minimum and appear dead cycle. It is a global optimization algorithm. The optimization object is a group of individuals, and the search process is carried out along multiple trajectories, which has implicit parallelism. However, the genetic algorithm has some disadvantages, such as poor local search ability and prone to premature phenomenon. Therefore, it is necessary to improve the algorithm to improve the efficiency and local search ability of the algorithm [12].

3.1. Improved Genetic Algorithm

3.1.1. Coding Method. Instead of using the commonly used binary coding, the binary coding of adjacent integers may have a large hamming distance. The real number coding method is adopted; that is, each gene value of an individual is represented by a real number within a specific range. This method is convenient to deal with the problems of multi-dimensional and high precision and improves the computational complexity and the computational efficiency of the algorithm.

3.1.2. Best Individual Retention. The basic method is to select in proportion according to the size of individual fitness. In this paper, the individuals with the highest fitness in the population are directly copied to the next generation, so that the individuals with the highest fitness; that is, the best individuals do not exchange and mutate in this generation [13]. This method speeds up the search speed and improves the local search ability.

3.1.3. Expected Value Selection Method. The expected value selection method can overcome the random error phenomenon that individuals with high fitness may be eliminated and individuals with low fitness may be selected [14–16] and improve the real-time performance of genetic algorithm. The specific steps are as follows:

1. Calculate the expected value of fitness:

   \[ \bar{f}_i = \frac{1}{n} \sum_{j=1}^{n} f_j, \]  

2. Calculate the expected value of individual survival:

   \[ \bar{R}_i = \frac{\bar{f}_i}{f_i}, \]  

3. \( \bar{R}_i \) is converted to integer \( R_i \) through the rounding principle. If, \( R_i = 0 \), the individual is eliminated

4. Mutation operator

It can restore the population diversity and ensure the global convergence of the algorithm. Therefore, according to the optimal solution retained by the previous generation [17], the variation probability is adjusted as follows:

\[ p_m = p_o (1 + \delta)^{H/C}, \]  

where \( p_o \) is the initial mutation probability, \( 0 < \delta < 1 \), \( \delta \) is the adjustment coefficient, \( H \) is the algebra maintained by the optimal solution, and \( C \) is a constant.

According to different fitness, the variation probability is adjusted as follows:

\[ p_m = p_o - \beta K, \]  

where \( \beta \) is the adjustment coefficient determined by the fitness value, \( 0 < \beta < 1 \), \( K \) is a constant, and \( K < P_o \).

Equation (3) ensures that when premature occurs and the optimal solution is maintained, the mutation probability increases until it jumps out of the local extreme point, while equation (4) reduces the mutation probability of individuals with large fitness [18], which improves the real-time performance of the algorithm.

3.2. Optimization Results. The EFT 20 kg industrial robot er20-c10 is selected, the parameters are input into the above formula, the appropriate interpolation points are selected, and the trajectory functions of 6 joints are calculated, and substituted into equations (3) and (4) to calculate the objective function. Taking the time interval as the variable and combined with the constraints, the results are optimized by improving the genetic algorithm. Firstly, local optimization and then global optimization improve the efficiency of the algorithm. The specific parameters are set as follows:

3.2.1. Local Optimization. Firstly, a small population of 30 chromosomes is selected, and the evolutionary algebra is set to 20 generations; so, a local optimization time can be obtained quickly.

3.2.2. Global Optimization. Then, expand the population to 100 chromosomes and increase the evolutionary algebra to 60 generations, so as to obtain a global optimization time [19]. The initial value of the time interval is 3 s, each joint trajectory is composed of 5 sections of B-spline curve, and the initial total energy consumption is 30247.56 J. The initial value is optimized for 8 times. The optimization results are shown in Table 1:

The constraint parameters of robot ER20-C10 are shown in Table 2:
According to Table 1, the result of the seventh optimization is the best, with a total time of 11.351 s and a total energy consumption of 28447.31 J. The optimal trajectory of the robot can be obtained by substituting the optimization result into the calculated trajectory function [20].

### 3.3. Joint Space Trajectory Planning

In order to meet the requirements of trajectory in joint space, the inverse kinematics problem is used to transform each path point into the joint angle value, and then the smooth function is fitted for each joint to make it pass through each path point in sequence from the starting point to the target point. For each path segment, each joint has a kind of running time to ensure that each joint reaches each path point [17]. Although each joint is the same in each path period, the fitting functions of all joints are independent. The trajectory planning in joint space does not require the trajectory shape as in right angle space, so, the calculation is simple and has no singularity. The trajectory planning of joint space needs to give the position and attitude of the starting point and the ending point under the condition of satisfying a certain constraint (for example, the displacement, velocity, and acceleration of each node are continuous) and interpolate each joint with different functions to obtain the motion trajectory of each joint. This section mainly discusses the application of interpolation method in joint trajectory [21].

If a single joint can move smoothly, \( \dot{\theta}(t) \) interpolation function needs to be bound by at least four conditions. There are angle and velocity constraints for both endpoints. The constraint of endpoint angle is the joint angle value of track initial point and track end point [22].

\[
\begin{align*}
\theta(0) &= \theta_0 \\
\dot{\theta}(t_f) &= \dot{\theta}_f
\end{align*}
\]  

(5)

In order to ensure continuous joint movement speed, it is specified that

\[
\begin{align*}
\dot{\theta}(0) &= 0 \\
\dot{\theta}(t_f) &= 0
\end{align*}
\]  

(6)

### 3.4. Rectangular Coordinate Space Straight Line Trajectory Planning

The spatial linear interpolation planning algorithm is a process of solving the position and attitude of each trajectory interpolation point when the position and attitude of the starting point and end point are known. Because in most cases, when the robot moves along a straight line, its end orientation attitude does not change; so, there is no need to interpolate the end attitude [23]. The position and pose of the two points \( P_1(x_1, y_1, z_1) \) and \( P_2(x_2, y_2, z_2) \) at the beginning and end of the spatial line relative to the global coordinates are known. Find the straight line length \( L \) from \( P_1 \) to \( P_2 \) as follows:

\[
L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.
\]  

(7)

It is specified that the speed when moving along a straight line is \( \Delta t \) and the interpolation time interval is \( \Delta t \), then the moving distance \( a \) in time \( \Delta t \) and the total number of interpolation \( N \) are

\[
d = v \cdot \Delta t, \\
N = \frac{L}{d} + 1.
\]  

(8)

(9)

Then, the displacement increment of each joint of the robot and the coordinate value of each interpolation point are

\[
\begin{align*}
\Delta x &= \frac{(x_2 - x_1)}{N} \\
\Delta y &= \frac{(y_2 - y_1)}{N} \\
\Delta z &= \frac{(z_2 - z_1)}{N}
\end{align*}
\]  

(10)

### 3.5. Spatial Arc Interpolation

In three-dimensional space, any three points (not collinear) can determine a circle in any plane. Spatial arc trajectory planning is processed through the following steps:

The first step is to find out the plane where the spatial circle is located and convert the circle in three-dimensional space to two-dimensional plane.

In the second step, the plane arc planning is used to interpolate the arc on the spatial plane.

The third step is to convert the points in the two-dimensional plane in the second step to the base coordinate system.

### 4. Verification and Implementation of Optimal Trajectory Planning of Industrial Robot

According to the robot 3D simulation environment and project requirements, the design objectives of robot trajectory planning are as follows: (1) verify the real-time and efficiency of the simulation platform, (2) verify and realize the
time optimal trajectory planning in the welding transition process of the robot to the bucket; and (3) verify and realize the optimal smooth trajectory planning of straight line and arc in the process of bucket welding. The time optimal and smooth optimal trajectory planning algorithm studied in the previous chapters of this paper has ensured the smoothness and efficiency of the trajectory planning goal of industrial robot. The next research content only needs to verify the trajectory planning problem in the welding process between robot and bucket [24, 25]. The optimal trajectory is obtained from the above optimization results, which is taken as the trajectory of the industrial robot model with degrees of freedom. The kinematic parameter curve is obtained by motion analysis in ADAMS to test whether it meets the constraints. If the constraints are satisfied, the result is optimal. If the constraints are not met, the optimization results will be reselected and simulated again until they meet the constraints. The trajectory planning process between industrial robot and 3D simulation environment is shown in Figure 2.
The simulated kinematic parameters include the rotation angle, angular velocity, and angular acceleration of the robot. The change curve during the movement is shown in Figures 3–4 below.

By analyzing Figures 3 and 4, it can be seen that the robot takes the optimized trajectory as the motion trajectory, and the six joints meet the kinematic constraints in the motion process [26, 27]. The maximum amplitude of each curve obtained by simulation is brought into the dynamic equation, and it can be verified that the six joints also meet the dynamic constraints in the process of motion.

The quantitative comparison of the obtained task planning results is shown in Table 3.

It can be seen from Figure 5 and Table 1 that the improved genetic algorithm can obtain better path combination planning results than the original method. In addition, from the perspective of the speed of evolutionary convergence, the evolutionary algebra required for the improved genetic algorithm to converge to the optimal average path length has been greatly reduced, which further proves the advantages of the method in improving industrial production efficiency by operating the path direction and sequence at the same time.

This paper sets up two scenarios: single workstation of handling robot and single workstation of welding robot. The time optimal trajectory planning algorithm was invoked in the single workstation of the transport robot [28, 29]. The transportation time was 15.034 s and 17.456 s, respectively, and the time was 2.422 s. The second-order trajectory planning algorithm with the best smoothness is called in the single workstation of the welding robot, and the robot welds the space straight line and the space curve, respectively, to the bucket. According to the difference between before and after time, the time for space linear welding is 5.462 s, and the time for space curve welding is 12.981 s. In order to compare with the quadratic smoothing optimization, only the time optimal trajectory planning algorithm is called in the single workstation of the welding robot to make the robot weld the bucket space curve weld. The time is 9.65 s, and the trajectory may produce small jitter, which should be avoided as much as possible in robot welding.
5. Conclusion

In this paper, the trajectory of industrial robot is obtained by quintic B-spline function, and the constraints such as speed, acceleration, acceleration, and torque are given. Then, the kinetic energy of the robot is taken as its target energy consumption function. The genetic algorithm is improved to improve the operation efficiency, local search ability, and real-time performance of the algorithm. The improved genetic algorithm is used to optimize the objective function and obtain the optimal result, so as to obtain the optimal trajectory. This trajectory is used as the motion trajectory of the robot simulation model. The kinematic and dynamic parameters of each joint of the robot are simulated and analyzed in ADAMS to meet the constraint conditions, and the rationality of the optimal trajectory is verified. The optimization strategy has strong practicability and can be used to optimize the trajectory of other types of industrial robots. The simulation results show that the trajectory planning algorithm in this paper can realize the short time and continuous smooth trajectory of the robot in the three-dimensional simulation of industrial robot and ensure the real-time, high efficiency, and smoothness of the robot trajectory in the simulation environment. The algorithms discussed in this paper are carried out in the off-line system, and the real-time trajectory tracking problem is not considered. How to apply the optimized robot trajectory to the practical application of the robot, the real-time tracking of the controller is a further problem to be studied.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that he/she has no conflicts of interest.

References

[1] X. Luo, S. Li, S. Liu, and G. Liu, “An optimal trajectory planning method for path tracking of industrial robots,” *Robotica*, vol. 37, no. 3, pp. 502–520, 2019.
[2] J. Ogbemhe, K. Mpofu, and N. Tlale, “Optimal trajectory scheme for robotic welding along complex joints using a hybrid multi-objective genetic algorithm,” *IEEE Access*, vol. 7, pp. 158753–158769, 2019.
[3] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, “Improved traffic sign detection and recognition algorithm for intelligent vehicles,” *Sensors*, vol. 19, no. 18, p. 4021, 2019.
[4] B. Zhou and Q. Wu, “A novel optimal method of robotic weld assembly line balancing problems with changeover times: a case study,” *Assembly Automation*, vol. 38, no. 4, pp. 376–386, 2018.
[5] D. Yang, Z. Liu, T. Shu, L. Yang, J. Ouyang, and Z. Shen, “An improved genetic algorithm for multiobjective optimization of helical coil electromagnetic launchers,” *IEEE Transactions on Plasma Science*, vol. 46, no. 1, pp. 127–133, 2017.
[6] S. Dian, H. Fang, T. Zhao, Q. Wu, and S. Li, “Modeling and trajectory tracking control for magnetic wheeled mobile robots based on improved dual-heuristic dynamic programming,” *Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1470–1482, 2020.
[7] J. Batista, D. Souza, J. Silva, K. Ramos, and A. Braga, “Trajectory planning using artificial potential fields with metaheuristics,” *IEEE Latin America Transactions*, vol. 18, no. 5, pp. 914–922, 2020.
A. Mohammad, J. Hong, and D. Wang, “Polishing of uneven surfaces using industrial robots based on neural network and genetic algorithm,” *International Journal of Advanced Manufacturing Technology*, vol. 93, no. 1, pp. 1–9, 2017.

C. Ding, L. Chen, and B. Zhong, “Exploration of intelligent computing based on improved hybrid genetic algorithm,” *Cluster Computing*, vol. 22, no. S4, pp. 9037–9045, 2019.

K. Ahmad, K. Khan, and A. Al-Fuqaha, “Intelligent fusion of deep features for improved waste classification,” *IEEE Access*, vol. 8, pp. 96495–96504, 2020.

W. Sun, Y. Huang, and H. Chen, “An improved adaptive genetic algorithm for optimizing pyroshock acceleration synthesis,” *IEEE Access*, vol. 7, pp. 132682–132693, 2019.

K. Ma, L. Han, X. Sun et al., “A path planning method of robotic belt grinding for workpieces with complex surfaces,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 728–738, 2020.

G. Luo, L. Zou, Z. Wang, C. Lv, and Y. Huang, “A novel kinematic parameters calibration method for industrial robot based on levenberg-marquardt and differential evolution hybrid algorithm,” *Robotics and Computer-Integrated Manufacturing*, vol. 71, no. 1, p. 102165, 2021.

Z. Wang, “Design and implementation of polishing and polishing flexible workstation based on industrial robot,” *International Core Journal of Engineering*, vol. 6, no. 4, pp. 193–197, 2020.

Z. G. Wu, C. Y. Lin, H. W. Chang, and P. T. Lin, “Inline inspection with an industrial robot (iiir) for mass-customization production line,” *Sensors*, vol. 20, no. 11, p. 3008, 2020.

M. Li, Z. Du, X. Ma, K. Gao, and Y. Gao, “System design and monitoring method of robot grinding for friction stir weld seam,” *Applied Sciences*, vol. 10, no. 8, p. 2903, 2020.

M. Lin, C. Yang, D. Li, and G. Zhou, “Intelligent filter-based slam for mobile robots with improved localization performance,” *IEEE Access*, vol. 7, pp. 113284–113297, 2019.

Y. Shen, “Improved chaos genetic algorithm based state of charge determination for lithium batteries in electric vehicles,” *Energy*, vol. 152, pp. 576–585, 2018.

Y. Juan and K. Seok-Bum, “Improved license plate localisation algorithm based on morphological operations,” *IET Intelligent Transport Systems*, vol. 12, no. 6, pp. 542–549, 2018.

H. Shan, L. Hongjing, W. Linlin et al., “Economic optimisation of microgrid based on improved quantum genetic algorithm,” *The Journal of Engineering*, vol. 2019, no. 16, pp. 1167–1174, 2019.

R. Zhang and J. Tao, “A nonlinear fuzzy neural network modeling approach using improved genetic algorithm,” *Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5882–5892, 2017.

Q. Ren, M. Li, and S. Han, “Discrimination and comparison experiments of basalt tectonic setting based on improved genetic algorithm-optimized neural network,” *Earth Science Frontiers*, vol. 26, no. 4, pp. 117–124, 2019.

G. S. Yadav and A. Ojha, “Improved security in the genetic algorithm-based image steganography scheme using hilbert space-filling curve,” *The imaging science journal*, vol. 67, no. 3, pp. 148–158, 2019.

T. Iida, D. Anzai, and J. Wang, “A three-dimensional embedded implant device localization method improved by genetic algorithm,” *International Journal of Wireless Information Networks*, vol. 24, no. 2, pp. 180–188, 2017.

L. Du, S. Bo, Y. Liu, F. E. Alsaadi, and A. Alsaedi, “Genetic algorithm-based compliant robot path planning: an improved bi-rrt-based initialization method,” *Assembly Automation*, vol. 37, no. 3, pp. 261–270, 2017.

Y. Wu and Q. Song, “Improved particle swarm optimization algorithm in power system network reconfiguration,” *Mathematical Problems in Engineering*, vol. 2021, 10 pages, 2021.

L. Jie, W. Liu, Z. Sun, and S. Teng, “Hybrid fuzzy clustering methods based on improved self-adaptive cellular genetic algorithm and optimal-selection-based fuzzy c-means,” *Neurocomputing*, vol. 249, pp. 140–156, 2017.

X. Guo, J. Huang, H. Liu, and Y. Chen, “Efficient p-cycle combination protection strategy based on improved genetic algorithm in elastic optical networks,” *IET Optoelectronics*, vol. 12, no. 2, pp. 73–79, 2018.

R. Ponciroli, N. E. Stauff, J. Ramsey, F. Ganda, and R. B. Vilim, “An improved genetic algorithm approach to the unit commitment/economic dispatch problem,” *Transactions on Power Systems*, vol. 35, no. 5, pp. 4005–4013, 2020.