

Research Article

Optimal Trajectory Planning of Grinding Robot Based on Improved Whale Optimization Algorithm

Ting Wang , Zhijie Xin, Hongbin Miao, Huang Zhang, Zhenya Chen, and Yunfei Du

School of Mechanical Engineering, North University of China, Taiyuan 030051, China

Correspondence should be addressed to Ting Wang; 2319263087@qq.com

Received 30 May 2020; Accepted 31 July 2020; Published 19 August 2020

Academic Editor: Andras Szekrenyes

Copyright © 2020 Ting Wang et al. 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.

Robot will be used in the grinding industry widely to liberate human beings from harsh environments. In the grinding process, optimal trajectory planning will not only improve the processing quality but also improve the machining efficiency. The aims of this study are to propose a new algorithm and verify its efficiency in achieving the optimal trajectory planning of the grinding robot. An objective function has been defined terms of both time and jerk. Improved whale optimization algorithm (IWOA) is proposed based on whale optimization algorithm (WOA) and differential evolution algorithm (DE). Mutation operation and selection operation of DE are imitated in the part of initialization to process the population initialized by WOA, and then, the search tasks of WOA are performed. Motion with a constant velocity of the end-effector is considered during fine grinding. The continuity of acceleration and velocity will be achieved by minimizing jerk, and at the same time, smooth robot movement can be obtained. Cubic spline interpolation is implemented. A six-axis industrial robot is used for this research. Results show that optimal trajectory planning based on IWOA is more efficient than others. This method presented in this paper may have some indirect significance in robot business.

1. Introduction

The optimal trajectory planning of a robot is based on the optimization of parameters and the improvement of methods. The most significant optimization criteria are the minimum execute time, minimum energy, and minimum jerk. Trajectory planning in Cartesian space is intuitive, and the movement trajectory and attitude of the end-effector are easy to observe. But, this method did not consider problems caused by kinematic singularities [1–4]. Trajectory planning in joint space can avoid those problems. The joint trajectories are obtained by interpolating functions which satisfy kinematic and dynamic constraints. The trajectory based on a system controller is easier to adjust in joint space [5–7].

A time-optimal trajectory planning method with constraints of velocity, acceleration, and jerk is presented in [8]. This optimization method adopts a hybrid optimization strategy and solves the problem from path description, kinematics of the robot, and constraints. The resulting trajectory is optimal relative to time, but not smooth enough.

An analytic method is used to obtain the solution of the minimum jerk trajectory of a point-to-point path. A minimum-time/jerk algorithm is presented, and it is verified by experiments in [9]. A direct analytical method is proposed in [10] to achieve the linear transformation of the industrial robot's characteristic equation, which is useful for industrial robot trajectory planning and control. According to [11], time-optimal trajectory planning can be considered as a nonconvex problem. Previous robot trajectory planning mainly focused on time optimization [12, 13]. In the recent years, multiobjective optimization under kinematic constraint has been under attention so as to improve the efficiency and quality and save energy. Meanwhile, many algorithms used to achieve optimal trajectory planning of robot are emerged. In order to get the global minimum jerk for trajectory planning of a 2-dof planar robot, a new method using GA is proposed in [14]. Aiming at trajectory planning of a free-floating robot, a jerk optimal trajectory planning algorithm based on particle swarm optimization algorithm is proposed in [15]. Joint Angle trajectory is

smooth and continuous, and the driving torque of the robot joint angle is minimized while the attitude adjustment of the free-floating space robot is completed. To obtain the optimal solution for the trajectory of the space robot, a multi-objective particle swarm optimization method is used in [16]. Invasive Weed Optimization algorithm is used to research the optimal solution for a robot arm in [17]. We consider general asymmetric bounds on velocity, acceleration, and jerk [18] and present a complete trajectory generation algorithm. A multiobjective differential evolution algorithm proposed in [19] is used in trajectory planning of a mobile robot. A multiobjective teaching learning-based optimization method proposed in [20] is used to achieve optimal trajectory planning of a welding robot. In [21], a combination of the modified potential field and numerical approach-based Jacobian to achieve path planning of a redundant robot is proposed. An improved B-spline curve is proposed, which made the angular velocity and angular acceleration smooth and continuous during trajectory movement, and the path of the robot end-effector in the Cartesian space is improved while guaranteeing the smoothness of joint movement [22]. Cubic spline is used to interpolate trajectories to ensure the continuity of jerk in [14]. Based on improved cuckoo search algorithm, a 3-5-3 polynomial interpolation trajectory planning algorithm is proposed in [23]. Aiming at an autonomous operation of the mobile system, Pirník et al. [24] present a designed algorithm, which is able to integrate inertial sensor data and can improve the trajectory planning efficiency.

The optimization methods proposed by different scholars are applied in specific scenes. In this paper, IWOA is used in grinding robot optimal trajectory planning with certain objective function, and the optimization results are interpolated by cubic spline. The paper is organized as follows: in Section 2, an optimization problem is introduced and objective function is defined. Section 3 describes the details of WOA and IWOA. In Section 4, grinding path and constraints of kinematic are given. Section 5 shows the results of some simulation and comparison of the jerk and work time before and after optimization. In Section 6, the conclusion is given, and the reasons of optimization results are discussed.

2. Define Objective Function

The smoothness of robot motion plays an important role in robot grinding because it will help reduce vibration and improve the grinding quality. In order to grind the workpiece surface effectively and obtain uniform material removal, it is necessary to plan each point on the grinding path. Better continuity of speed and acceleration can prevent the mechanical body from shock, overshoot, and oscillation [25]. By reducing the jerk, the stability of the robot can improve, and the continuity of speed and acceleration can be achieved. But, the execution time will increase significantly and will affect the production beat. Therefore, trajectory planning is needed to meet the requirements of minimum time and minimum jerk in some extent by introducing weight parameters K_T and K_J . Firstly, the grinding path is

defined in the Cartesian space, and then, the inverse kinematics is solved for the ABB IRB2600 robot to obtain the angles of each joint. Considering the constraints of speed, acceleration, and jerk, the objective function is defined as follows:

$$\min F = K_T \sum_{i=1}^{n-1} l_i + K_J \sum_{j=1}^M \sum_{i=1}^{n-1} \sqrt{\frac{1}{l_i} \int_{t=0}^{t_w} \left(P_j'''(t) \right)^2 dt}, \quad (1)$$

with

$$\begin{aligned} |P_j'(t)| &\leq Vm_j, \\ |P_j''(t)| &\leq Am_j, \\ |P_j'''(t)| &\leq Jm_j, \end{aligned} \quad (2)$$

where $j = 1, 2, \dots, M$ is the number of joints and $i = 1, 2, \dots, n$ is the number of points in the grinding path. The more the points extracted on the curve, the closer it is to the contour curve of the workpiece. But, too many path points will affect the grinding efficiency of the robot, so defining an appropriate number of path points is important. K_T and K_J are weight coefficients. l_i is the computational time interval between two continuous points on the joint path. t_w is the whole grinding time. $|P_j'(t)|$, $|P_j''(t)|$, and $|P_j'''(t)|$ are velocity, acceleration, and jerk of the j th joint, respectively. Vm_j , Am_j , and Jm_j are the upper bounds of velocity, acceleration, and jerk of the j th joint, respectively.

The first part of (1) represents the running time of the robot, and the second part is an expression about jerk. Jerk is vectors, so to introduce it into the objective function, taking its square for operation, cubic spline interpolation is selected.

$$l_i = \max_j \frac{|P_{j,i+1} - P_{j,i}|}{P^i}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, M, \quad (3)$$

$$P''' = \frac{P''_{j,i+1} - P''_{j,i}}{l_i}, \quad (4)$$

where $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, M$. According to the positions in joint space, time interval, and the constraints of the velocity and acceleration, the robot acceleration matrix is obtained and used to solve (3).

3. Improved WOA

3.1. WOA. WOA is an efficiency nature-inspired meta-heuristic optimization algorithm, which mimics the social behavior of humpback whales. The efficiency of the WOA is evaluated by solving 29 mathematical optimization problems and six structural optimization problems, and optimization results demonstrate that WOA is very competitive compared to the state-of-the-art optimization methods [26]. WOA is used to solve many industrial problems and illustrated in the literature [27–29].

Step 1: the population is initialized

Step 2: the fitness value of each search agent is calculated, and the current best individual is selected

Step 3: the strategy of shrinking circle and up with spiral-shape is selected

Step 4: it is checked whether each search agent exceeds the search boundary, the fitness value of each search agent is calculated, and the optimal position is updated

Step 5: the maximum iteration is reached, and the final results are obtained

Details about Step 3 are as follows:

3.1.1. Exploitation. Humpback whales move around their prey in a shrinking circle, while following a spiraling path. To model this behavior, we assume that there is a 50% chance of choosing between shrinking encircling mechanism and a spiral model to update the whale's position. The mathematical model is as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D}, & \text{if } p < 0.5, \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), & \text{if } p \geq 0.5, \end{cases} \quad (5)$$

where p is a random number in $[0,1]$, t is the current iteration, \vec{A} and \vec{D} are coefficient vectors, \vec{X}^* is the position of the best solution so far, and \vec{X} is the position vector. X^* is a better solution that will be updated in each iteration. $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ indicates the distance of the i th whale to the prey, b is a constant for defining the shape of logarithmic spiral, and l is a random number in $[-1, 1]$.

3.1.2. Encircling Prey. WOA assumes that the current solution is the target prey. First, the best search agent is defined, and then, the other search agents will attempt to update their positions towards the best search agent. This behavior is expressed in the following equations:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right|, \quad (6)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}. \quad (7)$$

The vectors \vec{A} and \vec{C} are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \quad (8)$$

$$\vec{C} = 2 \cdot \vec{r}, \quad (9)$$

where \vec{a} is linearly decreased from 2 to 0 and \vec{r} is a random vector in $[0, 1]$.

3.1.3. Prey Search. Humpback whales search randomly based on each other's position. Therefore, a random value greater than 1 or less than -1 is used to force the search agent away from a reference individual. The position of the search agent is updated according to the random selected search

agent. The mechanism and the $|\vec{A}| > 1$ allow WOA achieve global search. The mathematical model is as follows:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X} \right|, \quad (10)$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}} - \vec{A} \cdot \vec{D}, \quad (11)$$

where \vec{X}_{rand} is a random position vector chosen from the current population.

3.2. IWOA. DE has been proved to be an effective global search algorithm in solving complex global optimization problems. DE and WOA are combined to propose a new algorithm (IWOA), which has better ability of global optimization and great robust. Figure 1 is P-code of IWOA.

By imitating the operation of mutation and selection in DE, the initialize program of WOA is improved. This new initialize program makes population more representative and easier to obtain a global optimal solution. Population size is considered as the number of whales, G_m is the maximum number of iterations, mutation factor F is calculated by (11), and F_0 is the initial value of F :

$$F = F_0 \cdot 2 \exp\left(\frac{1 - G_m}{G_m}\right). \quad (12)$$

3.2.1. Mutation Operation. For the whale position $X_{h,G}$ ($h = 1, 2, \dots, N$), a new position $V_{h,G+1}$ can be generated by the following formula:

$$V_{h,G+1} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}), \quad (13)$$

where N is the number of whales and $r1$ and $r2$ are different integers randomly selected in $[1, N]$ and instead of index h . The mutation factor is a real constant on $[0, 2]$, and the amplification of differential vector $X_{r2,G} - X_{r3,G}$ is controlled.

3.2.2. Selection Operation. In order to determine whether vector $V_{h,G+1}$ can become new positions for whales in the iteration of $G+1$, it will be compared with $X_{h,G}$, and if the fitness value of the former is better than the latter, $X_{h,G}$ will be replaced by $V_{h,G+1}$ in the $G+1$ iteration; otherwise, $X_{h,G}$ will be retained. Then, positions of whales are updated.

4. Implement

In the process of robot fine grinding, keeping the grinding speed as a constant contributes to control the material removal rate, so as to effectively control the grinding quality. Therefore, the grinding speed is introduced as the intermediate variable and optimized by IWOA. Then, the optimal motion parameters of each joint of the robot are obtained. The constraints of speed, acceleration, and jerk are shown in Table 1.

The first grinding path is divided into 50 trajectory points, as shown in Figure 2.

```

Initialize the whales population  $X_i$  ( $i = 1, 2, \dots, n$ )
Perform mutation operations by the eq. (12)
Perform selection operations
Update the whales population  $X_i'$ 
Calculate the fitness of each search agent
 $X^*$  = the best search agent
while ( $t <$  maximum number of iterations)
  for each search agent
    Update  $a$ ,  $A$ ,  $C$ ,  $I$ , and  $p$ 
    if1 ( $p < 0.5$ )
      if2 ( $|A| < 1$ )
        Update the position of the current search agent by the eq. (6)
      else if2 ( $|A| \geq 1$ )
        Select a random search agent ( $X_{rand}$ )
        Update the position of the current search agent by the eq. (10)
      end if2
    else if1 ( $p \geq 0.5$ )
      Update the position of the current search by the eq. (4)
    end if1
  end for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent
  Update  $X^*$  if there is a better solution
   $t = t + 1$ 
end while
return  $X^*$ 

```

FIGURE 1: Pseudocode of IWOA.

TABLE 1: Kinematic limits of a six-axis robot.

Joint	Velocity (deg/s)	Acceleration (deg/s ²)	Jerk (deg/s ³)
1	175	100	120
2	175	100	90
3	175	100	125
4	360	100	200
5	360	100	180
6	500	100	230

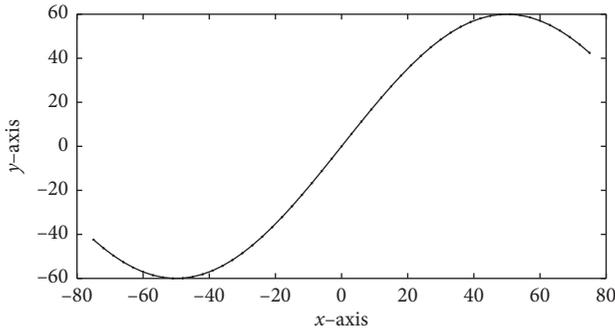


FIGURE 2: Defined trajectory points in the Cartesian coordinate system.

Then, the joint displacements of six axis for the ABB IRB2600 grinding robot are obtained by inverse kinematics solving. The value of K_T and K_J are chosen after trial of varying the values of each from 0 to 1 and set as 0.4 and 0.6, respectively.

The joint positions are used as the input of the IWOA, and MATLAB 2018b software is used to complete the optimization process. ABB RobotStudio software is used for simulation.

5. Results

The jerk values before and after optimization with respect to the intermittent points on the grinding path are illustrated in Figures 3(a)–3(f). In Figure 3(g), every joint's position and orientation are given. The comparison of whole grinding time before and after optimization with four kinds of algorithm is shown in Table 2.

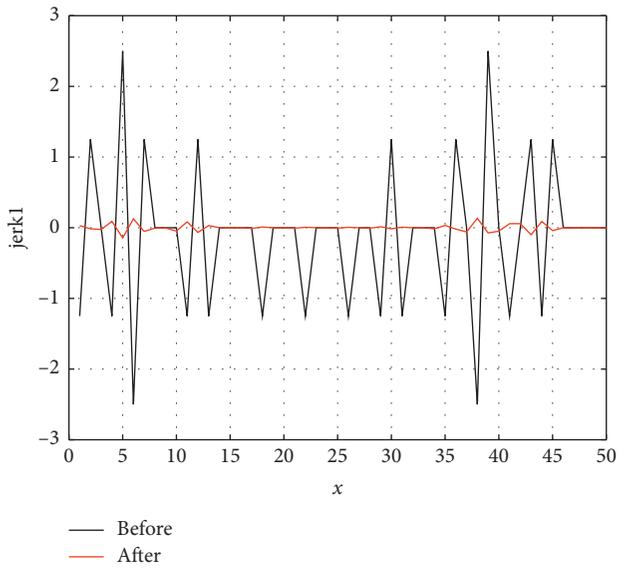
From Figure 3, it can be observed that jerk after optimization is significantly reduced compared with jerk before optimization and fluctuated around a constant value. The maximum value of jerk before optimization for every joint is 2.50 deg/s³, 2.50 deg/s³, 2.50 deg/s³, 250 deg/s³, 3.75 deg/s³, and 33.75 deg/s³. The maximum value of jerk after optimization for every joint is 0.144 deg/s³, 0.034 deg/s³, 0.066 deg/s³, 4.136 deg/s³, 0.039 deg/s³, and 0.042 deg/s³. No matter before or after optimization, the fluctuation range of joint 4 is significantly larger than others.

According to Table 2, it can be observed that grinding time after optimization is increased. The growing rate by using GA is 111.95%, using DE is 95.97%, using WOA is 71.26%, and using IWOA is 68.67%. Also, the growing rate is the smallest by using IWOA.

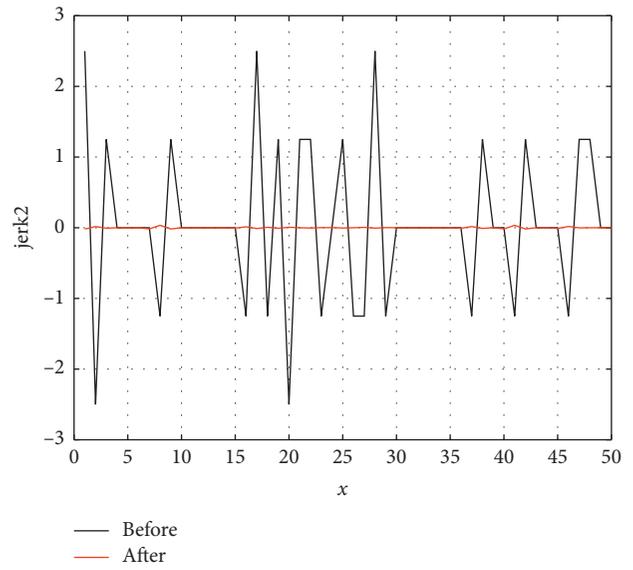
6. Discussion and Conclusions

In this paper, the objective function including an expression of jerk and time of total work for every joint IWOA is proposed by combining DE and WOA, and IWOA is used to get optimal trajectory of the grinding robot. After optimization, the goal of jerk being smaller is achieved and time is not increased significantly.

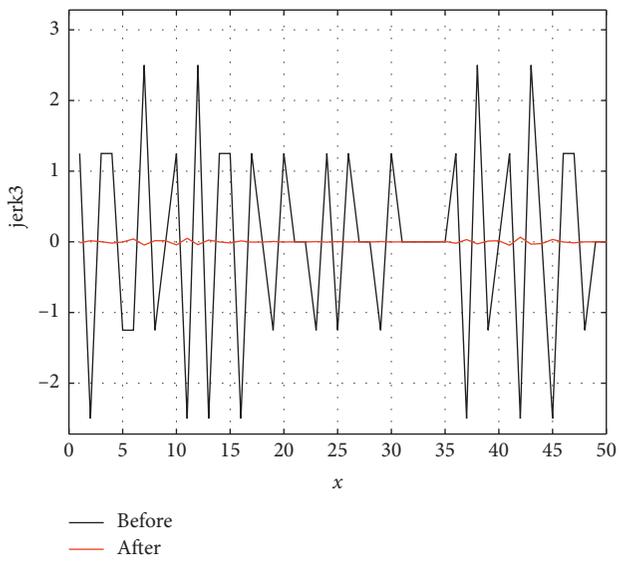
Jerk is reduced significantly by using IWOA. After optimization, jerk of every joint has reduced significantly. By reducing jerk, the movement path is smoother and the speed



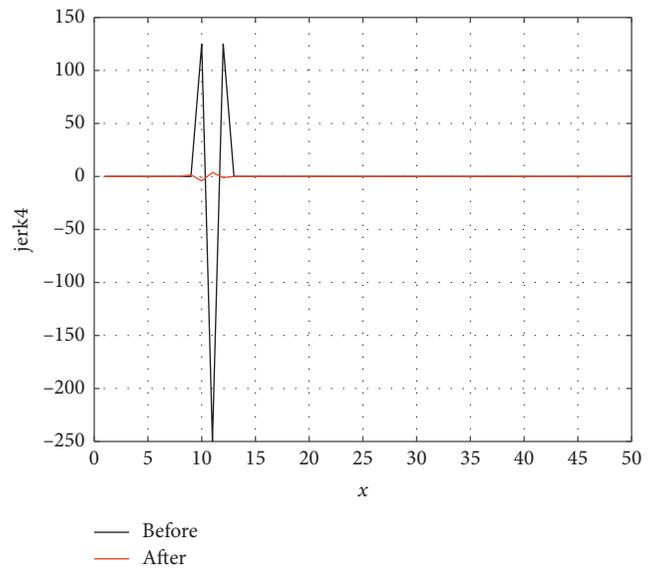
(a)



(b)



(c)



(d)

FIGURE 3: Continued.

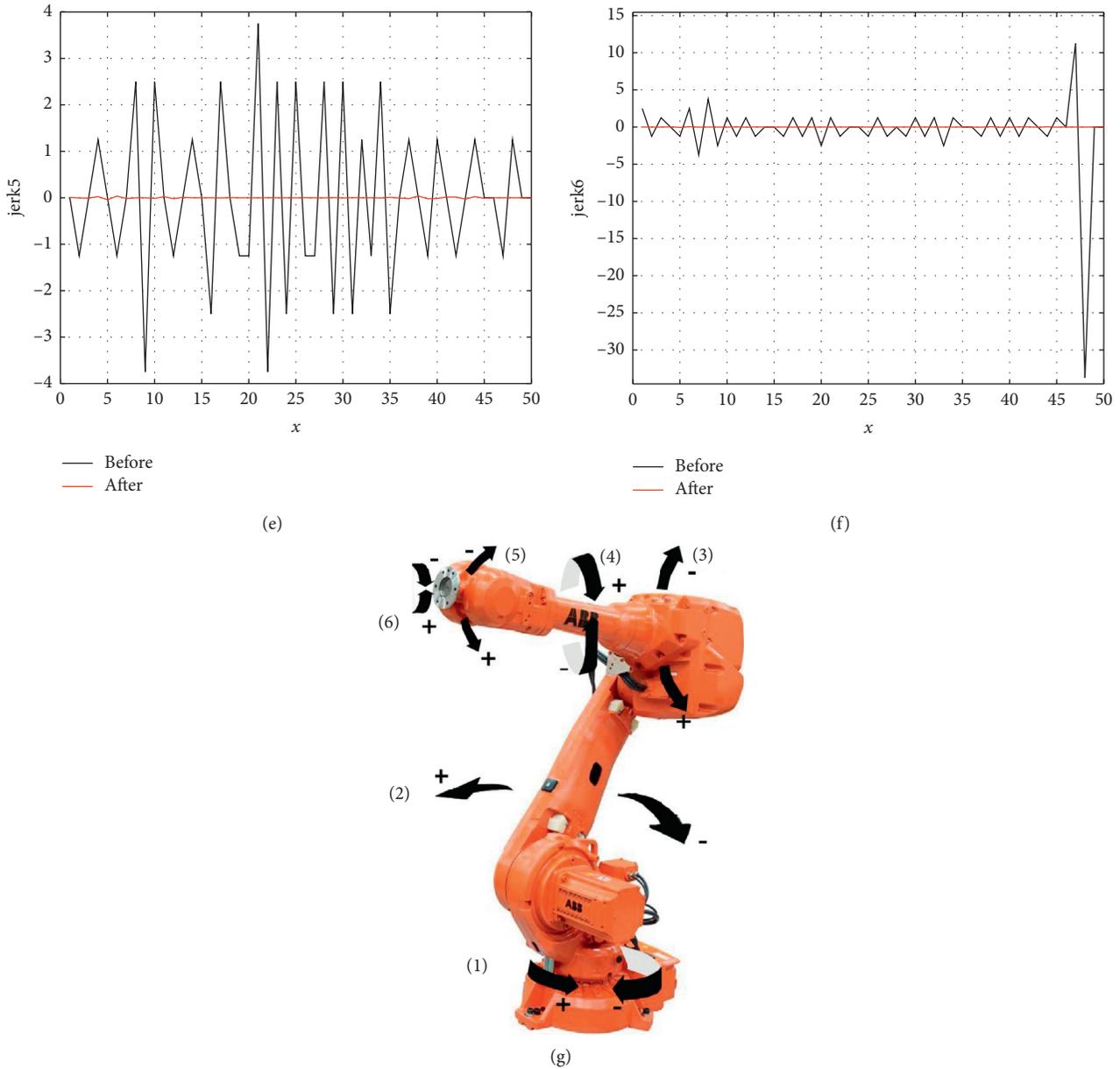


FIGURE 3: Jerk values of joints with respect to each trajectory point before and after optimization. (a) Jerk values of joint 1, (b) jerk values of joint 2, (c) jerk values of joint 3, (d) jerk values of joint 4, (e) jerk values of joint 5, (f) jerk values of joint 6, and (g) joints of the robot.

TABLE 2: The comparison of whole grinding time before and after optimization with algorithms.

Algorithm	Grinding time before (s)	Grinding time after (s)	Growing rate (%)
GA	20.08	42.56	111.95
DE	20.08	39.55	95.97
WOA	20.08	34.39	71.26
IWOA	20.08	33.87	68.67

and acceleration are more continuous, which can improve the grinding quality to some extent. The jerk fluctuation range of joint 4 is significantly larger than that of others because joint 4 has the biggest range of motion so as to ensure the end-effector passes every point on the grinding path.

IWOA is effective in performing an optimal trajectory planning of a robot. Compared with other three algorithms, the growing rate of time is the smallest by using IWOA. The reason is IWOA performs mutation and selection operation for the population initialized by WOA. Mutation and selection operation of IWOA mimic the operation of DE.

Then, the initial population becomes more representative, and the algorithm is more robust.

This new method of optimal trajectory planning based on IWOA will be useful in the industry of robot grinding, for example, the grinding of welding line for automobile and its accessories and burrs for hardware products.

Data Availability

All data are defined according to the needs of research and results are achieved by software.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was funded by the Central Government Guided Local Science and Technology Development Projects of China under grant no. YDZX20191400002765 and Key Research and Development Projects of Shanxi Province under grant no. 201903D421006.

References

- [1] C. Zheng, Y. Su, and P. C. Müller, "Simple online smooth trajectory generations for industrial systems," *Mechatronics*, vol. 19, no. 4, pp. 571–576, 2009.
- [2] S. Zhang, A. Maria Zanchettin, R. Villa, and S. Dai, "Real-time trajectory planning based on joint-decoupled optimization in human-robot interaction," *Mechanism and Machine Theory*, vol. 144, 2020.
- [3] G. Xuan and Y. Shao, "Reverse-driving trajectory planning and simulation of joint robot," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 384–388, 2018.
- [4] S. Chiaverini, B. Siciliano, and O. Egeland, "Experimental results on controlling a 6-DOF robot manipulator in the neighborhood of kinematic singularities," in *Experimental Robotics III. Lecture Notes in Control and Information Sciences*, T. Yoshikawa and F. Miyazaki, Eds., vol. 200, pp. 1–13, Springer, Berlin, Germany, 1994.
- [5] H. Wang, H. Wang, J. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematic constraints based on high-order polynomial curve," *Mechanism and Machine Theory*, vol. 139, pp. 284–293, 2019.
- [6] G. Chen, B. Yuan, Q. Jia, Y. Fu, and J. Tan, "Trajectory optimization for inhibiting the joint parameter jump of a space manipulator with a load-carrying task," *Mechanism and Machine Theory*, vol. 140, pp. 59–82, 2019.
- [7] X. Liu, C. Qiu, Q. Zeng, and A. Li, "Kinematics Analysis and Trajectory Planning of collaborative welding robot with multiple manipulators," *Procedia CIRP*, vol. 81, pp. 1034–1039, 2019.
- [8] J. Dong, P. M. Ferreira, and J. A. Stori, "Feed-rate optimization with jerk constraints for generating minimum-time trajectories," *International Journal of Machine Tools & Manufacture*, vol. 47, no. 12–13, pp. 1941–1955, 2007.
- [9] V. Zanutto, A. Gasparetto, A. Lanzutti, P. Boscariol, and R. Vidoni, "Experimental validation of minimum time-jerk algorithms for industrial robots," *Journal of Intelligent & Robotic Systems*, vol. 64, no. 2, pp. 197–219, 2011.
- [10] P. Božek, Ž. Ivandić, L. Alexander, V. Lyalin, and V. Tarasov, "Solutions to the characteristic equation for industrial robot's elliptic trajectories," *Tehnički Vjesnik*, vol. 23, no. 4, pp. 1017–1023, 2016.
- [11] A. Pallešchi, R. Mengacci, F. Angelini et al., "Time-optimal trajectory planning for flexible joint robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 938–945, 2020.
- [12] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 309–317, 2013.
- [13] F. J. Abu-Dakka, I. F. Assad, R. M. Alkhdour, and M. Abderahim, "Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 1–4, pp. 389–406, 2017.
- [14] P. Aurelio and V. Antonio, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, pp. 140–149, 2000.
- [15] Q. Hu, Y. Wang, and Z. Shi, "Minimum torque trajectory planning algorithm for free-floating space robot," *Journal of Harbin Institute of Technology*, vol. 43, no. 11, pp. 20–24, 2011.
- [16] P. Huang, K. Chen, and J. Yuan, "Motion trajectory planning of space manipulator for joint jerk minimization," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*, pp. 3543–3548, New York, NY, USA, 2007.
- [17] S. Abhronil, C. Tathagata, and K. Amit, "Energy efficient trajectory planning by a robot arm using invasive weed optimization technique," *Third World Congress on Nature And Biologically Inspired Computing*, vol. 43, pp. 311–316, 2011.
- [18] S. Daniel and D. Kevin, "Smooth cubic polynomial trajectories for human-robot interactions," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 3–4, pp. 851–869, 2019.
- [19] V. Sathiyi and M. Chinnadurai, "Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning," *Robotica*, vol. 37, no. 8, pp. 1363–1382, 2019.
- [20] R. Amruta, B. Deepak, and B. Biswal Bibhuti, "Optimal trajectory generation of an industrial welding robot with kinematic and dynamic constraints," *Industrial Robot-The International Journal of Robotics Research and Application*, vol. 47, no. 1, pp. 68–75, 2020.
- [21] S.-O. Park, M. C. Lee, and J. Kim, "Trajectory planning with collision avoidance for redundant robots using jacobian and artificial potential field-based real-time inverse kinematics," *International Journal of Control, Automation and Systems*, vol. 18, no. 8, pp. 2095–2107, 2020.
- [22] Y. Li, T. Huang, and D. G. Chetwynd, "An approach for smooth trajectory planning of high-speed pick-and-place parallel robots using quintic B-splines," *Mechanism and Machine Theory*, vol. 126, pp. 479–490, 2018.
- [23] W. Wang, Q. Tao, Y. Cao, X. Wang, and X. Zhang, "Robot time-optimal trajectory planning based on improved cuckoo search algorithm," *IEEE Access*, vol. 8, pp. 86923–86933, 2020.
- [24] R. Pirník, M. Hrubaš, D. Nemeč, T. Mravec, and P. Božek, "Integration of inertial sensor data into control of the mobile platform," in *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*, Springer International Publishing, Berlin, Germany, pp. 271–282, 2015.
- [25] Y. Sun, *Study on the Key Problems of Solid Rocket Engines Inner-Wall Insulation Grinding Robot*, Northeastern University, Boston, MA, USA, 2009.

- [26] M. Seyedali and L. Andrew, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [27] M. A. Elhosseini, A. Y. Haikal, M. Badawy, and N. Khashan, "Biped robot stability based on an A-C parametric Whale Optimization Algorithm," *Journal of Computational Science*, vol. 31, pp. 17–32, 2019.
- [28] H. Chen, Y. Xu, M. Wang, and X. Zhao, "A balanced whale optimization algorithm for constrained engineering design problems," *Applied Mathematical Modelling*, vol. 71, pp. 45–59, 2019.
- [29] M. M. Ahmed, A. A. Mahmoud, and Y. A. Almoataz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system," *Ain Shams Engineering Journal*, vol. 71, pp. 1–13, 2019.