Robot Time-Optimal Trajectory Planning Based on Improved Cuckoo Search Algorithm

WENJIE WANG, QING TAO, YUTING CAO, XIAOHUA WANG, AND XU ZHANG

College of Electronics and Information, Xi’an Polytechnic University, Xi’an 710048, China

Corresponding author: Xiaohua Wang (wangxiaohua@xpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Program 51905405, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Program 2019JQ-855, in part by the Natural Science Special Project of Education Department in Shaanxi Province of China under Program 19JK0375, and in part by the Graduate Innovation Fund Project under Grant chx2019026.

ABSTRACT The trajectory planning time is uncertain when there is no velocity constraint in the joint space of an operating robot. This paper proposes a 3-5-3 polynomial interpolation trajectory planning algorithm based on improved cuckoo search algorithm (ICS) which functions under a velocity constraint. Kinematics analysis of the robot is conducted and the 3-5-3 polynomial interpolation function is established as the foundation for trajectory planning. The size of the step size control factor is set as a variant which varies with the number of iterations, which accelerates the algorithm’s convergence and prevents premature falling into the local optimal solution, both problems in the traditional cuckoo search algorithm where the step size control factor is fixed. The proposed algorithm was used to optimize the trajectory of a test robot with interpolation time as the search space for optimization. Any interpolation time that did not meet the speed constraint was eliminated to secure the shortest possible robot running time. The UR robot was taken as a research object to simulate the joint motion trajectory planning in MATLAB. The proposed algorithm performed well and better realized the time-optimal trajectory under the constraint of speed compared to the traditional cuckoo search algorithm, particle swarm optimization algorithm, and genetic algorithm.

INDEX TERMS Improved cuckoo search algorithm, time-optimal, robot, trajectory planning, polynomial interpolation.

I. INTRODUCTION

Trajectory planning is the basis of robot motion control. Existing trajectory planning algorithms are developing towards intelligence as our technological environment grows continually sophisticated. A high-performance trajectory planning algorithm makes the robot run more smoothly, work more efficiently, and function over a longer service life. Trajectory planning algorithms are a hot topic in the robotics field.

Cubic polynomial interpolation trajectory planning is widely used due to its simplicity in calculation but may produce mechanical vibrations which increase joint abrasion and degrade control accuracy. Improving the interpolation order of the polynomial and deploying piecewise interpolation methods smooth the trajectory. Xu et al. [1] proposed the 3-5-3 spline function method for smoothed robot trajectory planning; this was a valuable contribution to the literature, but the scholars did not consider interpolation time. This time variable represents the robot’s ability to complete a specified motion as quickly as possible under multiple constraints. Trajectory planning based on polynomial interpolation is difficult to optimize under traditional methodology due to its high order and lack of convex hull. Using intelligent algorithms to achieve time optimal trajectory planning is a worthwhile research objective to this effect. Commonly used intelligent algorithms include the genetic algorithm [2], [3], particle swarm optimization [4], and convex optimization algorithm [5]. These methods require many iterations and complicated parameter adjustments; they show room for improvement compared to the cuckoo search algorithm [6] in terms of performance.

Nima and Farnaz [7] applied a cuckoo search algorithm in the cloud computing context to solve task scheduling problems. Prases and Dayal [8] used a cuckoo search algorithm for optimal path planning and obstacle avoidance in a mobile robot. Medjahed et al. [9] applied a cuckoo search algorithm to improve traditional hyperspectral image
classification techniques. Nie et al. [10] established a multi-skill human resource scheduling model, where a cuckoo search algorithm was used to allocate labor force time and resources. He et al. [11] used a cuckoo search algorithm to optimize a semi-supervised SVM for oil layer identification, which improved recognition rates and shortened recognition time by comparison to other methods. Yang et al. [12] applied a cuckoo search algorithm in a Kriging model to optimize fixture positioning layouts.

The cuckoo search algorithm step size control factor is a fixed constant, which causes the algorithm to converge slowly and fall into local optimal solution prematurely. Wang et al. [13] adopted an improved cuckoo search algorithm to solve the flow-shop scheduling problem, where a Nawaz-escort-ham (NEH) method was used to population initialization and a generalized opposition learning method was used to accelerate the convergence rate; this approach is effective but yields a highly complex algorithm. Ma et al. [14], Yang and Deb [15] proposed a dynamic adaptive cuckoo search algorithm which uses an adaptive step control to increase the cooperation among subgroups, improve the convergence speed, and increase the optimization accuracy.

There have been many other valuable contributions to the literature. Xia et al. [16] proposed a piecewise continuous function to limit the speed and acceleration in a robot trajectory planning process and did succeed in truncating the running time compared to traditional methods. However, the calculation was overly complex and computationally burdensome. He et al. [17] adopted an immune algorithm to simultaneously optimize the time and energy of a robot, but optimizing multiple goals at the same time is time-consuming. Chen et al. [18] used a harmony search algorithm to optimize a five-degree b-spline curve, and obtained a smooth and time-optimal trajectory, but the trajectory was not sufficiently smooth compared to the piecewise polynomial interpolation function. Verscheure et al. [19] adopted a convex optimization method to transform the time-optimal path tracking problem into a convex optimal control problem, which is effective but also costly in terms of the complexity of the solution process. Wang et al. [20] proposes a new algorithm named trajectory-planning beetle swarm optimization (TPBSO) algorithm for solving trajectory planning of robots. Kim et al. [21] presented a method for online trajectory planning that is computationally efficient, time-optimal, and maintains path following integrity.

This paper proposes a time-optimal 3-5-3 polynomial interpolation trajectory planning algorithm based on the cuckoo search algorithm under a velocity constraint. The UR robot was taken as the research object and the polynomial interpolation function was determined based on robot kinematics analysis. The improved cuckoo search algorithm adjusts the size of the control factor as iterations progress rather than using a fixed control factor. The improved cuckoo search algorithm can optimize interpolation time under the speed constraint and determine the quickest trajectory of the robot accordingly. Thus, the time uncertainty problem can be solved under a series of constraints (e.g., position, velocity, and acceleration). The proposed algorithm was validated in a series of simulations and by comparison against other state-of-the-art algorithms, as discussed in detail below.

II. ROBOT KINEMATICS ANALYSIS

Kinematics analysis is the basis of trajectory planning. When the robot is working, it is necessary to obtain the displacement of each joint based on the position and posture of tool at the end of the robot so as to achieve the specified position. The connecting rod coordinate system of the UR robot tested in this study was established according to the rules of the connecting rod coordinate system [22], as shown in Figure 1, with the D-H parameters shown in Table 1.
The transformation matrix between \( i \)th and \( i - 1 \)th links of robot \( i^{-1}T \) is:
\[
0T = \begin{bmatrix}
\cos \theta_i & -\sin \theta_i & 0 & 0 \\
\sin \theta_i & \cos \theta_i & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (1)

According to the given D-H parameters, the transformation matrix \( 0T \) of the first link can be obtained as follows:
\[
0T = \begin{bmatrix}
\cos \theta_0 & -\sin \theta_0 & 0 & 0 \\
\sin \theta_0 & \cos \theta_0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (2)

\( 1T, \frac{1}{2}T, \frac{3}{2}T, \frac{5}{2}T, \frac{7}{2}T \) can be obtained similarly to \( T \). The transformation matrix of each link can then be multiplied in turn to obtain the transformation matrix between the end of the robot. The base coordinates \( 0T \) can be obtained by multiplying the transformation matrix of each link in turn.

\[
0T = 1T\frac{1}{2}T\frac{3}{2}T\frac{5}{2}T\frac{7}{2}T
\] (3)

The mechanism of UR robot satisfies the Pieper criterion [23]: Three consecutive revolute joint axes intersect at a common point or three consecutive revolute joint axes are parallel. So its inverse kinematics solution has a numerical solution. The forward kinematics solution can be used to obtain the inverse kinematics solution analytically.

The solutions of joint angles \( \theta_1, \theta_2, \theta_5, \theta_6 \) can be obtained as follows.

Equation (3) can be rewritten as follows:
\[
0T - 0T = 1T\frac{1}{2}T\frac{3}{2}T\frac{5}{2}T\frac{7}{2}T
\] (4)

Because the elements of the third row and fourth column on the left and right sides of Equation (4) are equal to each other, the following formula holds:
\[
\theta_1 = A \tan 2(m, n) - A \tan 2(d_4, k)
\]
where:
\[
m = d_6a_y - p_y, \\
n = d_6a_x - p_x, \\
k = \pm \sqrt{m^2 + n^2 - d_4^2}
\]

The elements of the third row and third column on the left and right sides of Equation (4) are also equal to each other, so
\[
\theta_5 = \pm \arccos(a_xs_1 - a_ys_1)
\]
where \( s_1 = \sin \theta_1, \ c_1 = \cos \theta_1 \). Further, the elements of the third row and first column on the left and right sides of Equation (4) are equal to each other, so
\[
\theta_6 = A \tan 2(x/s_5, y/s_5)
\]

where \( x = n_x - n_xc_1, y = o_x - o_xc_1, s_5 = \sin \theta_5 \).

The solutions of joint angles \( \theta_3, \theta_2, \theta_4 \) can be obtained as follows. Equation (3) can be rewritten as:
\[
0T - 0T = 1T\frac{1}{2}T\frac{3}{2}T\frac{5}{2}T\frac{7}{2}T
\] (5)

The following relationships can be obtained because the right and left sides are equal (corresponding to elements in Formula (5)):
\[
\theta_3 = \pm \arccos \left(\frac{m^2 + n^2 - a_2^2 - a_3^2}{2a_2a_3}\right)
\]
\[
\theta_2 = A \tan 2(\sin \theta_2, \cos \theta_2), \quad \theta_4 = A \tan 2(s_6(n_xc_1 + n xs_1), c_6c_5 + n c_5c_6)
\]

where
\[
m_1 = a_3 \cos (\theta_2 + \theta_5) + a_2 \cos \theta_2, \\
n_1 = a_3 \sin (\theta_2 + \theta_5) + a_2 \sin \theta_2, \\
s_6 = \sin \theta_6, \quad c_6 = \cos \theta_6.
\]

The analytic expressions of the inverse kinematic solutions are then given.

### III. POLYNOMIAL INTERPOLATION FUNCTION CONSTRUCTION

The commonly used cubic polynomial interpolation trajectory planning algorithm may produce mechanical vibrations, a simulation of cubic polynomial interpolation trajectory planning for robot was carried out as is shown in the Figure 2.
solution as per the starting point, dropping point, and end point positions corresponding to the polynomial interpolation function. This makes the robot end pass through the required path point to satisfy the given trajectory. The 3-5-3 polynomial interpolation method is adopted in this study as follows:

\[
\begin{align*}
  h_1(t) &= a_{i13}t^3 + a_{i12}t^2 + a_{i11}t + a_{i10} \\
  h_2(t) &= a_{i25}t^5 + a_{i24}t^4 + a_{i23}t^3 + a_{i22}t^2 + a_{i21}t + a_{i20} \\
  h_3(t) &= a_{i33}t^3 + a_{i32}t^2 + a_{i31}t + a_{i30}
\end{align*}
\]

(6)

where \(a_{imn}\) represents the \(n\)th unknown coefficient of the interpolation trajectory of the \(i\)th joint in the \(m\)th \((m = 1, 2, 3)\) polynomial trajectory segment; \(h_{im}(t)\) Represents the \(m\)th trajectory segment of the \(i\)th joint and \(t\) is the time variable.

In the trajectory planning process, it is necessary to satisfy constraints regarding the position, velocity, and acceleration of the starting point \(\theta_1\), lifting starting point \(\theta_2\), descending point \(\theta_3\), and end point \(\theta_4\). Constraint conditions include the following.

1) The positions of the four interpolation points are known;
2) The velocity and acceleration of the starting point are 0;
3) The velocity and acceleration of the end point are 0;
4) The velocity and acceleration of the starting point and descending point are continuous.

The unknown coefficient can be deduced according to Equations (7)-(9) under these constraints. Equation (7) is the matrix of the interpolation time \(t_{1i}, t_{2i}, t_{3i}\) of the \(i\)th joint, Equation (8) is the position matrix of the joint angle, and Equation (9) is the coefficient matrix.

\[
T = \begin{bmatrix} A & B & 0 \\ 0 & C & D \\ 0 & 0 & E \\ F & G & H \end{bmatrix}
\]

(7)

where \(A, B, C, D, E, F, G, H\) are:

\[
\begin{align*}
  A &= \begin{bmatrix} t_{11}^3 & t_{11}^2 & t_{11} & 1 \\ 3t_{11}^2 & 2t_{11} & 1 & 0 \\ 6t_{11} & 2 & 0 & 0 \end{bmatrix}, \\
  B &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix} \\
  C &= \begin{bmatrix} 5t_{12}^3 & 4t_{12}^2 & t_{12} & t_{12} & t_{12} & 1 \\ 20t_{12}^2 & 12t_{12} & 6t_{12} & 2 & 0 & 0 \end{bmatrix} \\
  D &= \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix} \\
  E &= \begin{bmatrix} t_{13}^3 & t_{13}^2 & t_{13} & 1 \\ 3t_{13}^2 & 2t_{13} & 1 & 0 \\ 6t_{13} & 2 & 0 & 0 \end{bmatrix}
\end{align*}
\]

IV. TIME-OPTIMAL PROBLEM BASED ON IMPROVED CUCKOO SEARCH ALGORITHM

A. TRADITIONAL CUCKOO SEARCH ALGORITHM

The Cuckoo Search (CS) is a natural meta-heuristic algorithm proposed by Yang and Deb in 2009 inspired by the avian breeding process [24]. The algorithm solves complex problems quickly and effectively by combining the Levy flight random search method. For a straightforward description of the standard CS, the following three idealized rules are introduced:

1) In a randomly selected nest, a cuckoo produces only one egg at a time;
2) The best nests with good eggs are preserved for the next generation;
3) The number of available nests is constant, and the probability of cuckoo eggs being found by the host is \(P_a\). In this case, the host can destroy the egg or abandon the old nest and build a new one.

On the basis of these three ideal states, the specific steps of cuckoo search algorithm are as follows.

1) The population size, discovery probability, and the termination conditions are initialized.
2) The objective function value of the initial population is evaluated to find the current global optimal nest.

3) Low-quality nests are weeded out according to the probability that their eggs will be found.

4) Low-quality (high detection probability) nests are replaced with new nests created by Levy’s flight:

\[ x_i^{t+1} = x_i^t + \alpha L(\lambda) \]  

(10)

where \( x_i^t \) represents the location of the \( i \)th nest in the \( t \)th generation, \( \alpha \) represents the step size control factor, and \( L(\lambda) \) represents Levy-flight random search paths.

5) The best nest will be found again in the updated nest data.

6) If the termination condition is satisfied, the best nest is the optimal solution found so far. Otherwise, return to Step 3.

### B. IMPROVED CUCKOO SEARCH ALGORITHM

Although CS algorithm has been applied and studied in many fields, it still has some defects. The Levy’s flight randomness is large, so the standard CS algorithm lacks an effective mechanism to enhance the search depth and the algorithm’s convergence accuracy is relatively low. The step size control factor of the CS algorithm is defined in Equation (10) as an artificial constant; if it is too large, the search value of the algorithm will yield a poor solution in the later stages of operation while slowing down its convergence rate. If it is too small, the algorithm may fall into the local optimal solution too early, i.e., perform poorly overall. In this study, the CS algorithm was improved by targeting the step size control factor \( \alpha \): the original (fixed) one was replaced by an adaptive one.

The value range of \( \alpha \) decreases in each evolutionary generation. In the early stages of the algorithm, large step size control factors can be used to quickly find the region of high-quality solutions in the global range. The step size control factor is then gradually reduced as the iterations progress while conducting a detailed search of the local high-quality solution region, which enhances the convergence speed and performance of the algorithm.

The proposed expression of step size control factor \( \alpha \) is:

\[
\alpha(t_{i+1}) = \begin{cases} 
\alpha_{\text{max}} \exp(-t_i/t_{\text{max}}) & T < 0.35 \\
\alpha(t_i) & 0.35 \leq T \leq 0.65 \\
\alpha_{\text{min}} \exp(-t_i/t_{\text{max}}) & T > 0.65 
\end{cases}
\]  

(11)

The value of \( \alpha_{\text{min}} \) was set as 0.01 [25] in this paper. In order to find the appropriate value of \( \alpha_{\text{max}} \), the parameter \( \alpha_{\text{max}} \) was set to 0.4, 0.5, 0.75 and 1 respectively, other parameters unchanged, the first joint optimization time was taken as the objective function. The first joint optimization time of different \( \alpha_{\text{max}} \) is shown in Table 2, when the test run 15 times.

We can find when \( \alpha_{\text{max}} = 0.5 \), variance value and the running time are minimal and the average value is acceptable, so we choose \( \alpha_{\text{max}} = 0.5 \) as the most appropriate value of parameter \( \alpha_{\text{max}} \).

| \( \alpha_{\text{max}} \) | 0.4 | 0.5 | 0.75 | 1  |
|-------------------|-----|-----|-----|----|
| Average value     | 7.3458 | 7.3462 | 7.3573 | 7.3602 |
| Variance value    | 0.0039 | 0.0026 | 0.0052 | 0.0054 |
| Running time/s    | 1.5162 | 1.5148 | 1.5198 | 1.5210 |

where \( T \) is the ratio between the current number of iterations and the total number of iterations, \( \alpha_{\text{min}} \) is the lower limit of step size control factor, \( t_{\text{max}} \) is the maximum number of iterations, and \( t_i \) is the number of current iterations.

In order to find the appropriate time segment of \( T \), Rastrigin Function was chanced as the testing function, generations is 300, and run 30 times. the testing results of different sections are shown in Table 3.

It can be seen that the solution accuracy is the highest when 0.35-065 is the segmentation boundary, and the difference in running time is not big, so 0.35-0.65 is chosen as the dividing line.

In the initial phase of the algorithm when \( T < 0.35 \), the global search should be performed with large strides. The algorithm may reach its best update state in the middle phase, when \( 0.35 \leq T \leq 0.65 \). At this time, it should further search in the region where the high-quality solution is located and strengthen the local refined search while \( \alpha \) has the same value. At the later stage, when \( T > 0.65 \), a smaller step length should be used to search for the optimal value.

If the unknown coefficient \( \alpha_{\text{lima}} \) is selected as the objective function to be optimized, the optimized dimension of the improved CS algorithm is 14. This makes finding the optimal value very difficult. Taking the three interpolation times \( t_1, t_2, \) and \( t_3 \) of the \( i \)th joint as the objective function greatly reduces the search dimensions and makes the computation simpler. The objective function to be optimized serves to minimize the running time of each joint under the constraint of velocity:

\[
f(t) = \min (t_1 + t_2 + t_3) \quad \text{max} \{v_i\} < v_{i_{\text{max}}}
\]  

(12)

where \( v_i \) is the real-time velocity of polynomial trajectory segment and \( v_{i_{\text{max}}} \) is the restricted velocity of joints.

The algorithm flow is shown as follows:

1) The population size, discovery probability, and the termination conditions are initialized.

2) The objective function value \( F_i = f(X_i) \) of the initial population is evaluated.

### TABLE 1. The first joint optimization time with different \( \alpha_{\text{max}} \).

| \( \alpha_{\text{max}} \) | 0.4 | 0.5 | 0.75 | 1  |
|-------------------|-----|-----|-----|----|
| Average value     | 7.3458 | 7.3462 | 7.3573 | 7.3602 |
| Variance value    | 0.0039 | 0.0026 | 0.0052 | 0.0054 |
| Running time/s    | 1.5162 | 1.5148 | 1.5198 | 1.5210 |

### TABLE 2. The testing results of different sections of \( T \).

| \( T \) time segment | Average value | Variance value | Running time/s |
|---------------------|--------------|---------------|---------------|
| 0.1-0.9             | 1.3860 \times 10^6 | 4.5319 \times 10^7 | 2.169077 |
| 0.2-0.8             | 9.4376 \times 10^6 | 7.2635 \times 10^4 | 2.220416 |
| 0.3-0.7             | 2.4352 \times 10^7 | 5.5051 \times 10^3 | 2.185184 |
| 0.35-0.65           | 1.8590 \times 10^7 | 3.6797 \times 10^7 | 2.175913 |
| 0.4-0.6             | 5.1956 \times 10^6 | 3.1763 \times 10^6 | 2.158980 |
| 0.5                 | 1.4099 \times 10^6 | 1.6122 \times 10^6 | 2.068300 |
3) Levy flight random walk mechanism is used to generate new nest locations $X_j$, based on Equations (10)-(11).

4) Evaluate the fitness value $F_j = f(X_j)$ of the new nest location.

   If $f(X_j) < f(X_i)$, replace the original location nest with a new nest location. Otherwise, keep it in the same place.

5) If the termination condition is satisfied, the best nest is the optimal solution found so far. Otherwise, return to Step 3.

C. TIME-OPTIMAL TRAJECTORY PLANNING BASED ON IMPROVED CS

The step-wise process of the time-optimal trajectory planning process for the robot’s $i$th joint based on the improved CS algorithm is as follows.

1) Initialize the population by determining the number $N$ of host nests and the probability $p_a$ of a cuckoo egg being found. The location of the nest is randomly generated in the time-interpolated search space.

2) Solve the unknown polynomial coefficient by plugging the $N$ groups of time variables $t_{i1}$, $t_{i2}$, and $t_{i3}$ into Equations (7)-(9).

3) Validate the constraint by plugging the polynomial coefficient (Step 2) and the corresponding time variable into the derivative of Equation (6) to obtain the real-time joint velocity. Judge whether the velocity conforms to the constraint of Equation (13).

4) Calculate the fitness value. Screen the results of Step 3. If the velocity value is greater than the constraint value, the nest will be considered low-quality and eliminated according to its probability of discovery. If the velocity value is less than the constraint value, Equation (12) is taken as the objective function and the improved CS algorithm is used for iterative operation over the shortest possible interpolation time.

5) Produce a new nest based on the improved CS algorithm iteration.

6) Judge the final conditions. If the termination condition is satisfied, the algorithm operation ends; otherwise, return to Step 2.

V. SIMULATION VERIFICATION AND RESULTS ANALYSIS

A. TIME-OPTIMAL TRAJECTORY PLANNING RESULTS

Take the polishing work of denim fabric polishing robot as an example, the speed limit of the denim polishing robot in the grinding process, the safety and efficiency of robot operation should be considered. The speed limit depends on the speed of polishing tools when polishing the fabric manually and convert this velocity to joint space velocity. Then we set the speed limit to $35^\circ/s$.

The position and posture information of the robot end is shown in Table 4; the first three values in the path point are position information and the last three are posture information. According to this information, interpolation points in Cartesian space can be converted into joint space angle interpolation points via inverse solution. The corresponding angles of each joint at each path point are shown in Table 5. Each interpolation time is artificially given with uncertainty for the traditional 3-5-3 polynomial interpolation, which affects the operation efficiency. The position, velocity, and acceleration curves of each joint motion with interpolation time of 5 s are shown in Figure 2.

As shown in Figure 3, 4 and 5, the constraint conditions are satisfied as the velocity and acceleration of both the starting point and the ending point are 0 and the velocity and acceleration of the starting and descending points are continuous. Safety, stability, efficiency, and other factors are considered comprehensively in this case. The maximum speed of each joint of the robot is $35^\circ/s$, i.e., has not reached to the maximum speed allowed; the robot in this case is inefficient.

The improved CS algorithm was used to solve the optimal time of the 3-5-3 interpolation polynomial method with the velocity constraint. The solving of Joint 1 optimal time is
reported here as an example. The initial population of the bird nest was set to 15 in this case, the probability of cuckoo eggs being found was 0.25, the search optimization was carried out in the 3D solution space, the initial particle position was any random number within 0.1 to 5, the value range of the step size control factor was 0.01-0.5, and the number of iterations was 100.

The proposed improved CS algorithm was compared against the CS algorithm, particle swarm optimization (PSO), and genetic algorithm (GA) to obtain the optimal particle evolution diagram shown in Figure 6. The GA optimization capability is poor compared with other three algorithms and it has a longer minimum running time. The PSO converges quickly, but its solution accuracy is poor compared with the ICS or CS. The CS converges after about 90 iterations, while the ICS converges after about 30 iterations; in effect, the improved CS algorithm proposed here has better solution accuracy and faster convergence speed than the other algorithms we tested.

Figure 7 shows a comparison of the running time of the four algorithms. The running environment in this case was Windows 764-bit, processor: Intel (R) core (TM) i5-3210M, CPU frequency: 2.50GHz Memory:4GB, Software: MATLAB2016b.

The four algorithms were run 15 times independently. The average running time of GA was 1.8791 s, that of the PSO was 2.8531 s, the CS was 1.5265 s, and the ICS was 1.4957 s, as is shown in Table 6.

The improved CS algorithm, in other words, very quickly solved the problem of the optimal 3-5-3 interpolation polynomial time under the velocity constraint. Then the improved CS algorithm was also used to obtain the optimal time of each segment of other joints. The results are shown in Table 7.
Since the joint motion of the robot has time synchronization, Max \( t_{i1} = 3.8190 \text{ s} \), Max \( t_{i2} = 2.9135 \text{ s} \), and Max \( t_{i3} = 3.3355 \text{ s} \) were selected as the maximum interpolation time of each segment. Compared with the unoptimized 3-5-3 interpolation polynomial, the interpolation time was reduced by 4.932 s.

Figure 8, 9 and 10 show the position, velocity, and acceleration curves of the robot as the improved CS algorithm obtains the optimal interpolation time. After optimization, the constraint conditions are satisfied wherein the velocity and acceleration of both the starting and ending points are 0; the velocity and acceleration of the starting and descending points are also continuous. The change of position of the unoptimized position curve is not much different from that of the optimized curve. The optimized joint speed has increased but not exceeded the limit speed compared to the nonoptimized speed curve, and tends toward the limited speed. This suggests that the proposed method is correct, but further, that it improves the robot’s work efficiency while ensuring safety.

Figure 11 shows the trajectory curve of the robot end with the optimal 3-5-3 polynomial interpolation optimized by the improved CS algorithm. The motion trajectory of the robot end passes through the given interpolation point smoothly, which further validates the proposed algorithm.

### B. ROBUSTNESS TEST RESULTS OF THE ICS

The improved cuckoo search algorithm presented in this paper can be used on different velocity and acceleration conditions by setting different limitations on velocity. as below:

Taking the motion of the first joint as an example, the solving ability of the algorithm under different speed constraints is verified. The Table 8 shows the optimization time and total time of each section obtained by using the improved cuckoo algorithm under different speed constraints. Figure 12 shows the change curves of position, speed and acceleration under different speed constraints. According to the figure, the algorithm can be used under different speed constraints.

| Speed constraint \( \left( \times \text{ s}^{-2} \right) \) | Optimal time of each segment (s) | Total time(s) |
|-----------------|-----------------|--------------|
| 20              | 4.9539 2.5020 5.2187 | 12.6746      |
| 40              | 2.4807 1.2405 2.6137 | 6.3349       |
| 60              | 1.6414 0.8442 1.7339 | 4.2195       |

In order to test the anti-interference ability of the ICS, taking the first joint as an example, add perturbations to the interpolation points, as shown in the following Table.

The total optimization time of the first joint is 7.4005s by using the improved cuckoo optimization while consider the disturbance, as is shown in Figure 13., the optimal time of
We can find there was only a slight change in the two situations. Then, combined the 3-5-3 polynomial interpolation the curves of position, velocity and acceleration can be obtained as shown in figure 15, 16 and 17. We can find that the speed curve does not exceed the speed limit. In addition, it is found that the algorithm has better anti-interference ability by testing other joints.
We will use the trajectory planning method proposed in this paper to test the performance of this method and lay a foundation for the actual product development in the next physical experiment of denim fabric polishing robot.

VI. CONCLUSION
This paper presents an improved CS algorithm where the step size control factor serves as a variable constant and changes with the number of iterations. This improves the convergence speed of the original cuckoo search algorithm and prevents falling into the local optimal solution prematurely, which is the case when the step size control factor is fixed.

The interpolation time of each segment is uncertain in the traditional 3-5-3 polynomial interpolation, which adversely affects the working efficiency, mechanical wear, and production safety of the robot in which it is applied. In this study, we developed a time-optimal trajectory planning method based on an improved CS algorithm in joint space with velocity as the constraint condition. Compared with CS, PSO, and GA techniques, the proposed algorithm performs better robot time optimal trajectory planning according to MATLAB simulations based on the UR robot as an example. After optimization via the proposed algorithm, each joint of the robot runs smoothly, and each joint satisfies the speed constraint. Its speed and acceleration are continuous and smooth at the interpolation time point, while the running time is greatly shortened, and operating efficiency is improved compared to other algorithms. And the improved CS algorithm has good anti-interference ability and the adaptability. These results indicate that the proposed algorithm has high practical value.

VII. FUTURE WORK
We will use the trajectory planning method proposed in this paper to test the performance of this method and lay a foundation for the actual product development in the next physical experiment of denim fabric polishing robot.

REFERENCES
[1] X. Xiang-Rong, W. Xiao-Gang, and Q. Feng, “Trajectory planning of robot manipulators by using spline function approach,” in Proc. 3rd World Congr. Intell. Control Autom., vol. 2, 2000, pp. 1215–1219.
[2] P. Hugo, D. Nalanda, R. Espi, V. Sethi, P. Pilidis, D. Zammitt-Mangion, J.-M. Rogero, and R. Entz, “Application of genetic algorithm for preliminary trajectory optimization,” SAE Int. J. Aerosp., vol. 4, no. 2, pp. 973–987, 2011.
[3] M. Cong, X. Xu, and P. Xu, “Time-jerk synthetic optimal trajectory planning of robot based on fuzzy genetic algorithm,” Int. J. Intell. Syst. Technol. Appl., vol. 8, nos. 1–4, pp. 185–199, 2010.
[4] P. Huang and Y. Xu, “PSO-based time-optimal trajectory planning for space robot with dynamic constraints,” in Proc. IEEE Int. Conf. Robot. Biomimetics (ROBO), Dec. 2006, pp. 1402–1407.
[5] Q. Zhang, S. Li, J.-X. Guo, and X.-S. Gao, “Time-optimal path tracking for robots under dynamics constraints based on convex optimization,” Robotica, vol. 34, no. 9, pp. 2116–2139, Sep. 2016.
[6] M. Jamil, H.-J. Zepernick, and X.-S. Yang, “Levy flight based cuckoo search algorithm for synthesizing cross-ambiguity functions,” in Proc. IEEE Int. Commun. Conf. (MILCOM), Nov. 2013, pp. 823–828.
[7] N. J. Navinipour and F. S. Milani, “Task scheduling in the cloud computing based on the cuckoo search algorithm,” Int. J. Model. Optim., vol. 5, no. 1, pp. 44–47, Feb. 2015.
[8] P. K. Mohanty and D. R. Parhi, “Optimal path planning for a mobile robot using cuckoo search algorithm,” J. Exp. Theor. Artif. Intell., vol. 28, nos. 1–2, pp. 35–52, Mar. 2016.
[9] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, “Binary cuckoo search algorithm for band selection in hyperspectral image classification,” IAENG Int. J. Comput. Sci., vol. 42, no. 3, pp. 1–9, 2015.
[10] H. Nie, B. Liu, F. Xie, Z. Liu, and H. Yang, “A cuckoo search algorithm for scheduling multiskilled workforce,” J. Netw., vol. 9, no. 5, pp. 1346–1353, 2014.
[11] Z. He, K. Xia, W. Niu, N. Aslam, and J. Hou, “Semi-supervised SVM based on cuckoo search algorithm and its application,” Math. Problems Eng., vol. 2018, pp. 1–13, Sep. 2018.
[12] B. Yang, Z. Wang, Y. Yang, Y. Kang, and Y. Li, “Optimum fixture locating layout for sheet metal part by integrating Kriging with cuckoo search algorithm,” Int. J. Adv. Manuf. Technol., vol. 91, nos. 1–4, pp. 327–340, Jul. 2017.
[13] H. Wang, W. Wang, H. Sun, Z. Cui, S. Rahnamayan, and S. Zeng, “A new cuckoo search algorithm with hybrid strategies for food shop scheduling problems,” Soft Comput., vol. 21, no. 15, pp. 4297–4307, Aug. 2017.
[14] H.-S. Mu, S.-X. Li, S.-F. Li, Z.-N. Li, and J.-S. Wang, “An improved dynamic self-adaption cuckoo search algorithm based on collaboration between subpopulations,” Neural Comput. Appl., vol. 31, no. 5, pp. 1375–1389, May 2019.
[15] X.-S. Yang and S. Deb, “Multiobjective cuckoo search for design optimization,” Comput. Oper. Res., vol. 40, no. 6, pp. 1616–1624, Jun. 2013.
[16] X. Zheng-Ya, Z. Zhou-Kang, Z. Cheng, and H. Liang, “A method for industrial robot manipulators trajectory planning based on piecewise continuous function,” in Proc. Int. Conf. Artif. Intell. Robot. Int. Conf. Automat., Control Robot. Eng. (ICAIR-CACRE), 2016, pp. 1–5.
[17] T. He, Y. Zhang, F. Sun, and X. Shi, “Immune optimization based multi-objective six-DOF trajectory planning for industrial robot manipulators,” in Proc. 12th World Congr. Intell. Control Automat. (WCICA), Jun. 2016, pp. 2945–2950.
[18] Y. Chen, L. Yan, H. Wei, and T. Wang, “Optimal trajectory planning for industrial robots using harmony search algorithm,” Ind. Robot, Int. J., vol. 40, no. 5, pp. 502–512, Aug. 2013.
[19] D. Vorschewre, B. Demuynckere, J. Swevers, J. De Schutter, and M. Diehl, “Time-optimal path tracking for robots: A convex optimization approach,” IEEE Trans. Autom. Control, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.
[20] L. Wang, Q. Wu, F. Lin, S. Li, and D. Chen, “A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators,” IEEE Access, vol. 7, pp. 154331–154345, 2019.
[21] J. Kim and E. A. Croft, “Online near time-optimal trajectory planning for industrial robots,” Robot. Comput.-Integr. Manuf., vol. 58, pp. 158–171, Aug. 2019.
[22] J. J. Craigh, Introduction to Robotics: Mechanics and Control, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.
[23] S. Bruno and O. Khatib, Springer Handbook of Robotics. Berlin, Germany: Springer, 2008, pp. 84–85.
[24] X. S. Yang and S. Deb, “Cuckoo search via Lévy flights,” in Proc. World Congr. Nature Biologically Inspired Comput. (NABIC), 2009, pp. 210–214.
[25] W. H. Yang, J. R. Liu, and Y. Zhang, “A new local-enhanced cuckoo search algorithm,” Int. J. Comput. Sci. Math., vol. 8, no. 2, pp. 175–182, 2017.

WENJIE WANG was born in Baoji, China, in 1988. He received the B.S. and Ph.D. degrees in mechanical engineering from the College of Mechanical and Electrical Engineering, Harbin Engineering University, in 2012 and 2018, respectively. He is currently a Distinguished Associate Professor of control science and engineering with Xi'an Polytechnic University. He has led and participated in over ten projects. He has authored over 20 articles and over ten inventions. His current research interests include mechatronics, surgical robot, parallel robot, and intelligent equipment.
QING TAO received the B.S. degree in automation from the Liren College, Yanshan University, Hebei, China, in 2017. He is currently pursuing the M.S. degree in control engineering with Xi’an Polytechnic University, China.

His research interests include robot dynamics, trajectory planning, and control system design.

YUTING CAO received the B.S. degree in automation from the Changzhou Institute of Technology, Changzhou, China, in 2017. She is currently pursing the M.S. degree in control engineering with Xi’an Polytechnic University, China.

Her research interests include robot dynamics, trajectory planning, and control system design.

XIAOHUA WANG was born in Baiquan, Heilongjiang, China, in 1972. She received the B.S. and M.S. degrees from Xi’an Polytechnic University, Xi’an, China, in 1997 and 1997, respectively, and the Ph.D. degree in mechanical engineering from the Xi’an University of Science and Technology, Xi’an, in 2010.

Since 1997, she has been a Teacher with Xi’an Polytechnic University. Since 2006, she has been a Professor with the College of Electrics and Information, Xi’an Polytechnic University. She is the author of four books and more than 40 articles. Her research interests include supervisory control of discrete event systems, system dynamics, and innovation linear and nonlinear systems.

XU ZHANG is currently pursuing the bachelor’s degree with Xi’an Polytechnic University, China.

His research interests include robot dynamics, trajectory planning, and control system design.