



# Time Optimal Trajectory Planning Based on Improved Sparrow Search Algorithm

Xiaofeng Zhang<sup>1</sup>, Fan Xiao<sup>1\*</sup>, XiLiang Tong<sup>1,2,3\*</sup>, Juntong Yun<sup>1,2</sup>, Ying Liu<sup>2,3</sup>, Ying Sun<sup>1,2,3\*</sup>, Bo Tao<sup>1,2,3</sup>, Jianyi Kong<sup>2,3,4</sup>, Manman Xu<sup>1,3,4</sup> and Baojia Chen<sup>5\*</sup>

<sup>1</sup>Key Laboratory of Metallurgical Equipment and Control Technology of Ministry of Education, Wuhan University of Science and Technology, Wuhan, China, <sup>2</sup>Research Center for Biomimetic Robot and Intelligent Measurement and Control, Wuhan University of Science and Technology, Wuhan, China, <sup>3</sup>Hubei Key Laboratory of Mechanical Transmission and Manufacturing Engineering, Wuhan University of Science and Technology, Wuhan, China, <sup>4</sup>Precision Manufacturing Research Institute, Wuhan University of Science and Technology, Wuhan, China, <sup>5</sup>Hubei Key Laboratory of Hydroelectric Machinery Design & Maintenance, China Three Gorges University, Yichang, China

## OPEN ACCESS

### Edited by:

Tinggui Chen,  
Zhejiang Gongshang University, China

### Reviewed by:

Tarik Rashid,  
University of Kurdistan Hewler, Iraq  
Shurong Li,  
Beijing University of Posts and  
Telecommunications (BUPT), China

### \*Correspondence:

Fan Xiao  
xiaofan@wust.edu.cn  
XiLiang Tong  
tongxiliang@wust.edu.cn  
Ying Sun  
sunying65@wust.edu.cn  
Baojia Chen  
cbjia@163.com

### Specialty section:

This article was submitted to  
Bionics and Biomimetics,  
a section of the journal  
Frontiers in Bioengineering and  
Biotechnology

**Received:** 11 January 2022

**Accepted:** 14 February 2022

**Published:** 22 March 2022

### Citation:

Zhang X, Xiao F, Tong X, Yun J, Liu Y,  
Sun Y, Tao B, Kong J, Xu M and  
Chen B (2022) Time Optimal Trajectory  
Planning Based on Improved Sparrow  
Search Algorithm.  
Front. Bioeng. Biotechnol. 10:852408.  
doi: 10.3389/fbioe.2022.852408

Complete trajectory planning includes path planning, inverse solution solving and trajectory optimization. In this paper, a highly smooth and time-saving approach to trajectory planning is obtained by improving the kinematic and optimization algorithms for the time-optimal trajectory planning problem. By partitioning the joint space, the paper obtains an inverse solution calculation based on the partitioning of the joint space, saving 40% of the inverse kinematics solution time. This means that a large number of computational resources can be saved in trajectory planning. In addition, an improved sparrow search algorithm (SSA) is proposed to complete the solution of the time-optimal trajectory. A Tent chaotic mapping was used to optimize the way of generating initial populations. The algorithm was further improved by combining it with an adaptive step factor. The experiments demonstrated the performance of the improved SSA. The robot's trajectory is further optimized in time by an improved sparrow search algorithm. Experimental results show that the method can improve convergence speed and global search capability and ensure smooth trajectories.

**Keywords:** trajectory planning, inverse kinematics, configuration space, time optimization, improved sparrow search algorithm

## INTRODUCTION

Robots are one of the vehicles for replacing some of the work of humans, combining the strengths of many disciplines and being used in a wide range of fields, such as mechanism (Yoshikawa, 1985; Park, 1995), computer vision (Huang et al., 2020; Bai et al., 2022), intelligent control (Yang et al., 2019; Yu et al., 2020; Yun et al., 2022a), artificial intelligence (Yu et al., 2019; Li et al., 2019; Sun et al., 2020a), signal processing (He et al., 2019; Jiang et al., 2019; Ma et al., 2020; Sun et al., 2020b; Weng et al., 2021), etc.. Time is the primary factor to be considered in engineering scenarios, and optimizing time is of great research significance for improving the efficiency of robotic arms. Time optimization aims to maximize the execution efficiency of the motor within the permissible limits. Simply increasing the torque and power of the motor is not feasible, as it results in greater energy consumption and inertia. Depending on the space in which the planning object is located, it can be divided into trajectory planning in Cartesian space and trajectory planning in joint space. When planning trajectories in Cartesian space, the position and pose of the robot arm end-effector (EE) are

intuitive (Kant and Zucker, 1986; Xu et al., 2009; Zheng et al., 2009), making it more suitable for scenarios with strict requirements on the trajectory movement process, such as machining (Sun et al., 2020c), welding (Hao and Wang, 2021), rehabilitation robot (Liu et al., 2016; Sun et al., 2018; Li et al., 2020; Luo et al., 2020), etc. However, inverse kinematic calculation takes up significant computational resources during the trajectory motion, and it is difficult to circumvent the kinematic singularity problem. When planning a trajectory in joint space (Chen and Zalzal 1997; Verscheure et al., 2009; Stilman, 2010), the object of planning is the joint angles corresponding to the target position, the whole process is done in joint space, and the motion is fully accessible, avoiding a large number of inverse kinematic calculations and eliminating the need to consider the problem of singularities, thus making it faster. After completing the planning in joint space, the trajectory needs to be mapped back to Cartesian space. The mapping process is highly non-linear, so the trajectory is unpredictable without constraints (Liu et al., 2013). However, many scenarios only require identifying several key points on the trajectory and do not require tracking the movement between these points, such as rigid body grasping (Miao et al., 2015; Hu et al., 2019; Jiang and Zheng, 2019; Duan and Sun, 2021; Jiang et al., 2021; Liu et al., 2022a), intelligent sorting, plant planting, etc. From a control point of view, the motor drive system acts directly on the joint angles of the axes so that the smoothness of the joint trajectory is more important than the smoothness of the trajectory in task space.

Time optimization is a single objective optimization problem for trajectory planning. The minimum time for the trajectory motion is calculated with the constraint that the velocity and acceleration do not exceed a set maximum value, provided that the trajectory and velocity are smoothly continuous and the acceleration is continuous (Gasparetto et al., 2012). The optimal time problem is generally solved by determining the interpolation function of the trajectory and then determining the coefficients of the interpolation function to ensure time is optimal. Polynomials are widely used in the choice of trajectory interpolation functions (Biagiotti and Melchiorri, 2008). The polynomials' order grows larger to obtain more accurate trajectories, which causes the Runge's phenomenon. Multi-segment polynomials are used in most scenarios in order to eliminate this effect. However, this brings about the creation of non-differentiable points within the interpolation function, which has a negative effect on the normal operation of the motor. The polynomial is at least third order derivable to ensure smoothness of velocity and continuity of acceleration. Higher orders provide more parameters and mean that the calculation becomes difficult. Alternatively, B splines have excellent performance in smoothing, and Thompson SE developed a method for constructing joint trajectories using B splines (Thompson and Patel, 1987). However, the properties of B splines dictate that the trajectories are fitted and not interpolated, and the absolute accuracy of the interpolated points cannot be guaranteed.

This work aims to reduce the time of robot trajectory planning in terms of both inverse solution calculation and trajectory

optimization. The algorithm used for trajectory optimization can be extended to other configurations of robots. This paper completes the trajectory planning of a UR5 collaborative robot (cobot) in joint space by means of cubic spline interpolation and obtains a time-optimal trajectory using an improved sparrow search algorithm (SSA). The complete flow of the trajectory motion is considered in this paper for the time optimization of the trajectory and the optimization of the inverse solution calculation method. The method of partitioning the joint space of the UR5 cobot is discussed, and one-to-one correspondence with the subspaces is obtained, significantly reducing the time required for the inverse kinematics (IK) calculation. The smoothness of the joint trajectory and the velocity is ensured by a cubic spline interpolation method. The problem of its tendency to fall into local optima is improved by improving the sparrow search algorithm to obtain a time-optimal trajectory with maximum velocity and acceleration as constraints.

This paper is organized as follows. **Section 2** details other studies relevant to the work in this paper. The mathematical model for time-optimal trajectory planning is presented in **Section 3**, and the problems and solutions in the various processes of trajectory planning are discussed in detail. The sparrow search algorithm and its improved methods are shown in detail in **Section 4**. **Section 5** shows the experimental results of the algorithm. Some outlooks and conclusions are given in **Section 6**.

## RELATED WORKS

The time-optimal problem of trajectories has been actively studied. Earlier proposed algorithms (Bobrow et al., 1985; Shin and McKay, 1985) are based on the position-phase plane. These algorithm's transform the time-optimal problem into a function of  $\theta$  and  $v$  as parameters to find the optimal problem. In short, for each point of the path, the minimum time to pass the entire path is obtained by passing at the maximum speed. The maximum velocity allowed for each point is found in the plane formed by  $\theta$  and  $v$ , and is made continuous when switching from point to point. These methods do not consider the acceleration continuity, which is not possible for an actuator moving in actual operation to produce discontinuous acceleration. This approach of ignoring actuator dynamics leads to two adverse effects: first, the discontinuous acceleration causes the actuator motion always to be delayed concerning the reference trajectory. This significantly reduces the tracking accuracy of the trajectory. In addition, constant switching can achieve discontinuous acceleration, but this introduces high-frequency oscillations to the actuator.

Solutions to these problems may be found in this literature (Shiller, 1996; Constantinescu, 1998; Constantinescu and Croft, 2000). In these methods higher order derivatives are added for finite control, which requires the establishment of third order dynamic equations. However, building accurate kinetic models is often challenging to accomplish.

Another approach that does not require a dynamics model is to use a smoothing function to express the trajectory in joint

space. The torque of the actuator is directly reflected in the joint variation of the robot so that a smooth trajectory will result in a smooth model. In the usual case, the spline interpolation function is widely used. Three constraints need to be considered after describing the joint trajectory using the spline function.

- 1) Speed limit;
- 2) Acceleration limit;
- 3) Jerk limit.

The third spline function has time as the horizontal axis and joints as the vertical axis, and the third-order derivability ensures that the acceleration is continuous. The calculation of the optimal trajectory time is completed by finding the minimum value of time under the satisfied constraints.

Tandem robots are generally made up of multiple joints, and the motion of each joint is coupled, making it challenging to complete the optimization process through numerical solutions. Swarm intelligence search algorithms have shown great vitality in such problems (Tian et al., 2020; Chen et al., 2021a; Chen et al., 2021b; Chen et al., 2022). Swarm search algorithms are widely used in robotics, such as inverse solution computation (Zhao et al., 2022), control (Liu G et al., 2021; Wu et al., 2022), pose recognition (Li et al., 2019a; Tao et al., 2022a) and other nonlinear problems (Huang et al., 2019; Sun et al., 2020d; Hao et al., 2022). Recently published optimisers (Ghafari and Gharehchopogh, 2012; Abedi and Gharehchopogh, 2020; Abdollahzadeh et al., 2021a; Gharehchopogh et al., 2021a; Abdollahzadeh et al., 2021b; Benyamin et al., 2021; Gharehchopogh et al., 2021b; Gharehchopogh and Abdollahzadeh, 2021; Goldanloo and Gharehchopogh, 2021; Mohammadzadeh and Gharehchopogh, 2021; Zaman and Gharehchopogh, 2021; Gharehchopogh, 2022) have achieved good performance but may not suit industrial scenarios with high real-time requirements. The particle swarm optimization algorithm (PSO) is used to search for the global time-optimal trajectory of a spatial robot in conjunction with robot dynamics (Huang and Xu, 2006). Huang uses a multi-objective particle swarm optimization algorithm optimization method for the multi-objective optimization of the motion trajectory of a space robot (Huang et al., 2008). Liu and Zhang (Zhang et al., 2018; Liu and Rodriguez, 2021) used a quintuple polynomial for trajectory planning for the PUMA560 robot and proposed an improved genetic algorithm (GA) to accomplish time-optimal trajectory planning.

These works demonstrate the feasibility of the group search algorithm for this problem, but accuracy and convergence speed remains problematic. The sparrow search algorithm is an algorithm proposed by Jiankai Xue (Xu et al., 2022) in 2020. The algorithm outperforms PSO, GA, grey wolf optimization algorithms (GWO). It is widely used other search algorithms on uni-modal and multi-modal test functions and is widely used in problems such as path planning for mobile robots (Liu et al., 2022b), control of photovoltaic microgrids (Yuan et al., 2021) and optimization of battery stack model parameters (Liu Y et al., 2022). We find that SSA is suitable for time-optimal trajectory planning problems and improves the initial population generation in the original algorithm through the Tent chaotic mapping method. An

adaptive step size factor adjusts the individual update to improve the global search capability. Time-optimal trajectory planning was completed on the UR5 collaborative robot, and experimental results demonstrate the effectiveness of the method.

A complete process includes trajectory determination, inverse solution solving, and trajectory optimization in practical motion. Depending on the scenario, the conditions for determining the trajectory are different. Obtaining information about the environment in these scenarios can be done by different sensors, such as myoelectric signals (Li et al., 2019b; Cheng et al., 2020; Cheng et al., 2021; Yang et al., 2021; Liu et al., 2022c), visual sensors (Jiang and Li, 2019; Jiang and Li, 2021; Tan et al., 2020; Huang et al., 2021; Liao et al., 2021; Liu X. et al., 2022; Sun et al., 2021, 2022; Yun et al., 2022b), multi-sensor fusion (Li et al., 2019c; Liao et al., 2020; Tao et al., 2022b), etc. The theoretical time required for the robot to complete the motion of a specified trajectory includes the motor execution time and the kinematic computation time. Therefore, trajectory planning is closely integrated with inverse kinematic solving. However, none of the above methods takes into account the time taken up in the trajectory planning by the inverse solution calculation. Based on the work described above, this paper also combines the unique domain theory to improve the computational efficiency of the algorithm further when the trajectory is in motion.

## TIME-OPTIMAL TRAJECTORY PLANNING

A complete trajectory planning process should include key point selection, inverse solution calculation, trajectory function setting and optimization. The coordinates of the key points on the trajectory are generally known in a Cartesian coordinate system. The inverse kinematics calculations allow the conversion of Cartesian coordinates into joint space, and the inverse kinematics of multi-joint robots is often multi-solvable. The UR5 cobot has eight sets of inverse solutions for the same pose. Assume that given a discrete sequence  $p_i$  of  $n$  interpolated sample points, the corresponding inverse solution of the joint angles is  $\mathbf{Q}_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{i8}] = IK(p_i)$ , where  $\theta_{ij}$  is a one-dimensional vector of six elements corresponding to the angle of each joint,  $i = 1, 2, \dots, n, j = 1, 2, \dots, 8$ . It is necessary to filter the inverse solution by a certain condition after the joint angle has been found, which often follows the principle of minimum joint variation between adjacent poses. As each joint has a different influence on the EE's trajectory, the amount of variation needs to be weighted in the calculation. The joint angle  $q_i$  corresponding to the  $i$ th sample point, i.e. the  $ind$ -th set of solutions in  $\mathbf{Q}_i$ , can be found from the following equation

$$(\Delta\theta, ind) = \min \left( \sum_{m=1}^6 w_m |\theta_{ij,m} - q_{i-1,m}| \right) \quad (1)$$

$$q_i = \mathbf{Q}_i(ind)$$

where  $q_{i-1}$  is the joint angle corresponding to the  $(i-1)$ -th sample point,  $m$  is the ordinal number of the joint angle,  $m = 1, 2, \dots, 6$ .  $w_m$  is the weighting factor corresponding to the  $m$ th joint,  $\Delta\theta$  is the minimum variation, and  $ind$  is the ordinal number of the minimum variation corresponding to in  $\mathbf{Q}_i$ .

The calculation and filtering of the inverse solution can occupy a lot of computational resources and waste much time, especially when the number of sample points is large. The literature (Wenger, 1992; Wenger, 2019) analyzed in depth the method of partitioning joint spaces and obtained the conclusion that the Jacobi matrix equation ( $\det(\mathbf{J}) = 0$ ) can partition the joint space into subspaces with the same number of solutions as the inverse, and these subspaces are called unique domains. However, they only provide a geometric analysis and do not suggest specific applications. The determinant of the Jacobi matrix for the UR5 cobot is

$$\begin{aligned} |\mathbf{J}| &= Pn_1 Pn_2 Pn_3 \\ Pn_1 &= a_2 a_3 s_5 \\ Pn_2 &= s_3 \\ Pn_3 &= a_2 c_2 + a_3 c_{23} - d_5 s_{234} \end{aligned} \quad (2)$$

where  $c$  represents the function  $\cos$ ,  $s$  denotes the function  $\sin$ , subscript  $i$  denotes  $\theta_i$ , subscript  $ij$  denotes  $\theta_i + \theta_j$ , subscript  $ijk$  denotes  $\theta_i + \theta_j + \theta_k$ . The rest of the paper uses expressions simplified in this way.

Each unique domain corresponds to a unique inverse resolution. It is shown that the choice of a suitable unique domain ensures the uniqueness of the inverse solution analytic, thus avoiding the problem of multiple solutions to the inverse kinematics. The pose  $\mathbf{T}$  of EE is already known in inverse kinematic computing, and the general form of  $\mathbf{T}$  can be expressed as follows

$$\mathbf{T} = \begin{bmatrix} \vec{n} & \vec{d} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where  $\vec{n}$  is the normal lapse,  $\vec{d}$  is the direction vector,  $\vec{a}$  is the approach vector, and  $\vec{p}$  is the position vector.

The unique domain corresponding to the inverse solution of the trajectory does not change when the robot's configuration does not change throughout its motion. Therefore only one unique domain corresponding to the inverse solution can be used when doing trajectory planning. The inverse solution has only one set of analytic solutions when the unique domain is determined (Liu Y et al., 2022). Combined with the method proposed by Xiao (Xiao et al., 2021), the calculation method of inverse kinematics of UR5 cobot is obtained. The IK analytical expression is

$$\begin{aligned} \theta_1 &= \text{atan2}\left(\frac{-d_4 p_x + A p_y}{-d_4^2 - A^2}, \frac{d_4 p_y + A p_x}{-d_4^2 - A^2}\right) \\ \theta_2 &= \text{atan2}\left(\frac{-M P_2 - N P_2}{-P_1^2 - P_2^2}, \frac{-M P_2 + M P_1}{-P_1^2 - P_2^2}\right) \\ \theta_3 &= \text{atan2}(s_3, c_3) \\ \theta_4 &= \text{atan2}(o_z s_5 + c_5 (n_z c_6 - a_z s_6), a_z c_6 + n_z s_6) - \theta_2 - \theta_3 \\ \theta_5 &= \text{atan2}\left(k_2 \sqrt{1 - (o_y c_1 - o_x s_1)^2}, o_y c_1 - o_x s_1\right) \\ \theta_6 &= \text{atan2}\left(\frac{(n_x s_1 - n_y c_1)}{s_5}, \frac{(a_x s_1 - a_y c_1)}{s_5}\right) \end{aligned} \quad (4)$$

where

**TABLE 1** | DH parameters of UR5 cobot, including link offset  $d_i$ , link length  $a_i$ , twist angle  $\alpha_i$  and joint angle  $\theta_i$ .

No	$d_i$ (m)	$a_i$ (m)	$\alpha_i$ (rad)	$\theta_i$ (rad)
1	0.1625	0	$\pi/2$	$\theta_1$
2	0	-0.425	0	$\theta_2$
3	0	-0.3922	0	$\theta_3$
4	0.1333	0	$\pi/2$	$\theta_4$
5	0.0997	0	$-\pi/2$	$\theta_5$
6	0.0996	0	0	$\theta_6$

$$\begin{aligned} A &= k_1 \sqrt{p_x^2 + p_y^2 - d_4^2} \\ c_3 &= N^2 + M^2 - a_2^2 - a_3^2 / 2a_2 a_3, \quad s_3 = k_3 \sqrt{1 - c_3^2} \\ N &= d_1 - p_z - d_5 c_{234}, \quad M = k_3 \sqrt{p_x^2 + p_y^2 - d_4^2} + d_5 s_{234} \\ P_1 &= a_3 s_3, \quad P_2 = a_2 + a_3 c_3 \end{aligned} \quad (5)$$

and  $a_2, a_3, d_4, d_5$  are the DH parameters of the robot and the exact values can be obtained in **Table 1**.  $k_1, k_2$  and  $k_3$  as 1 or  $-1$ .

The inverse solution calculation leads to values of  $k_1, k_2$ , and  $k_3$  that are not unique, which is why the joints have multiple solutions. There are eight sets of solutions in total. As the robot's configuration does not change when the trajectory is in motion, it is not necessary to consider all cases of  $k_1, k_2, k_3$ , and only a set of values needs to be chosen. According to the unique domain theory, we let  $k_1 = 1, k_2 = -1, k_3 = 1$ .

The inverse kinematic solution of the UR5 robot can be completed from **Eqs. 4** and **5**. The trajectory between sample points is generally done by interpolation. Assuming that the time corresponding to sample point  $p_j$  is  $t_j$ , the whole trajectory  $s(t)$  can be expressed as

$$\begin{aligned} s(t) &= \begin{cases} s_1(t), t \in [t_0, t_1] \\ s_2(t), t \in [t_1, t_2] \\ \vdots \\ s_n(t), t \in [t_{n-1}, t_n] \end{cases} \\ s_j(t) &= u_{k,j} t^k + u_{k-1,j} t^{k-1} + \dots + u_{1,j} t + u_{0,j} \end{aligned} \quad (6)$$

where  $u_{k,j}, u_{k-1,j}, \dots, u_{0,j}$  are constant coefficients and  $k$  is the order of  $s_j(t)$ ,  $j = 1, 2, \dots, n$ .

A spline curve is a special function defined by a polynomial segment. In engineering applications, spline interpolation is more reliable than polynomials. The interpolation error is small even if the order of the spline curve is not high, thus avoiding Runge's phenomenon. In order to balance the smoothness of the trajectory with the speed of the calculation, we use cubic splines, which guarantee the continuity of the acceleration. Thus the trajectory  $s_j(t)$  can be written

$$s_j(t) = u_{3,j} t^3 + u_{2,j} t^2 + u_{1,j} t + u_{0,j} \quad j = 1, 2, \dots, n \quad (7)$$

The value of  $s_j(t)$  at each interval endpoint is equal to the value at the sample point, i.e.  $s_j(t_{j-1}) = q_{j-1}$ ,  $s_j(t_j) = q_j$ . The derivatives of  $s_j(t)$  at the endpoints of each interval are continuous, i.e.  $s_j'(t_j - 0) = s_j'(t_j + 0)$ . The second-order derivatives of  $s_j(t)$  at the endpoints of each interval are continuous, i.e.  $s_j''(t_j - 0) = s_j''(t_j + 0)$ .

Noting that  $s_j''(t)$  is a first-order polynomial on the closed interval  $[t_{j-1}, t_j]$ , one can assume that the values of  $s_j''(t)$  at the endpoints of the interval are known, i.e.  $s_j''(t_{j-1}) = M_{j-1}$ ,  $s_j''(t_j) = M_j$ . Then

$$s_j''(t) = \frac{(t_j - t)M_{j-1} + (t - t_{j-1})M_j}{h_j} \tag{8}$$

where  $h_j = t_j - t_{j-1}$ .

By computing the integral of Eq. 8, we are able to obtain a general expression for the cubic spline at any moment t in the closed interval  $[t_{j-1}, t_j]$

$$s_j(t) = \frac{(t_j - t)^3 M_{j-1} + (t - t_{j-1})^3 M_j + (t_j - t)(6w_{j-1} - M_{j-1}h_j^2) + (t - t_{j-1})(6w_j - M_jh_j^2)}{6h_j} \tag{9}$$

From Eq. 8, we can obtain that there are (n+1) unknown variables in the whole trajectory  $s(t)$  ( $M_0, M_1, \dots, M_n$ ). In order to find out all the parameters, it is necessary to construct independent equations for  $M_0, M_1, \dots, M_n$  of independent equations

$$\mu_j M_{j-1} + 2M_j + \lambda_j M_{j+1} = \gamma_j \tag{10}$$

where  $\mu_j = h_j / (h_j + h_{j+1})$ ,  $\lambda_j = 1 - \mu_j$ ,  $\gamma_j = 6 [(w_{j+1} - w_j) / h_{j+1} - (w_j - w_{j-1}) / h_j] / (h_j + h_{j+1})$ .

For n cubic polynomials consisting of  $s(t)$ , there are (n-1) internal knots to obtain (n-1) equations in Eq. 8. In order to solve for the (n+1) unknown variables, we still need two other constraints. Noting that the velocity at the start and end of the whole trajectory is zero, i.e.  $s_j'(0) = 0$ ,  $s_j'(t_n) = 0$ . This leads to (n+1) independent equations

$$\begin{bmatrix} 2 & 1 & & & & & & & & & \\ \mu_1 & 2 & \lambda_1 & & & & & & & & \\ & \mu_2 & 2 & \lambda_2 & & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} & & & & & \\ & & & & 1 & 2 & & & & & \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-1} \\ \gamma_n \end{bmatrix} \tag{11}$$

where  $\mu_i, \lambda_i$  are known quantities and  $M_i$  are unknown variables.

The solution of Eq. 11 determines the display equation for the cubic polynomial, so the time-optimal problem can be described as solving for the minimum of the sum of  $h_j$  within the constraints, i.e.

$$\begin{aligned} f(t) &= \sum_{j=0}^n t_j \\ \text{s.t.} \quad &\begin{cases} \max(s'(t)) \leq V_{\max} \\ \max(s''(t)) \leq A_{\max} \\ \max(s'''(t)) \leq J_{\max} \end{cases} \end{aligned} \tag{12}$$

## IMPROVED SPARROW SEARCH ALGORITHM

### Sparrow Search Algorithm

The sparrow search algorithm models the behaviour of the sparrow as it forages for food and escapes predation. As with other heuristics, the sparrow population can be understood as a

randomly generated variable. The algorithm defines the different identities of individuals in the population: producers and joiners. The process of sparrow foraging is, in fact, an algorithmic search for optimal performance. According to a certain ratio, individuals with good energy are defined as producers during each foraging session and the remaining individuals as joiners. The joiner moves closer to the producer. The fitness function measures the magnitude of the energy. In addition, a certain percentage of individuals (typically 10–20% of the entire population) are randomly selected from the entire sparrow population to sense danger. These sparrows act like variant individuals in GA, preventing the population from falling into a local optimum. Their position shifts when certain conditions are triggered. This increases the diversity of the population and improves the possibility of individuals escaping the local optimum.

In the Tth foraging action, the producer's position is updated as follows

$$x_{i,j}^{T+1} = \begin{cases} x_{i,j}^T \exp\left(-\frac{i}{\alpha \cdot \text{iter}_{\max}}\right), & R_2 < ST \\ x_{i,j}^T + Q \cdot L, & R_2 \geq ST \end{cases} \tag{13}$$

where T is the current number of iterations, i is the sequence of individual sparrows in the population, and  $x_{i,j}$  denotes the coordinate position of the jth dimension of the current ith producer. Random factor  $\alpha \in (0,1]$ ,  $\text{iter}_{\max}$  represents the maximum number of iterations, Q is a random value that follows a normal distribution, L is a vector of the same dimension as the individual sparrow, random numbers  $R_2 \in [0,1]$ ,  $ST \in [0.5, 1]$ .

$R_2$  is a random number and ST is a constant in the domain of definition.  $R_2$  determines how the producer updates and this random setting allows for more variation in the producer's position, increasing the possibility of optimization seeking. For joiners

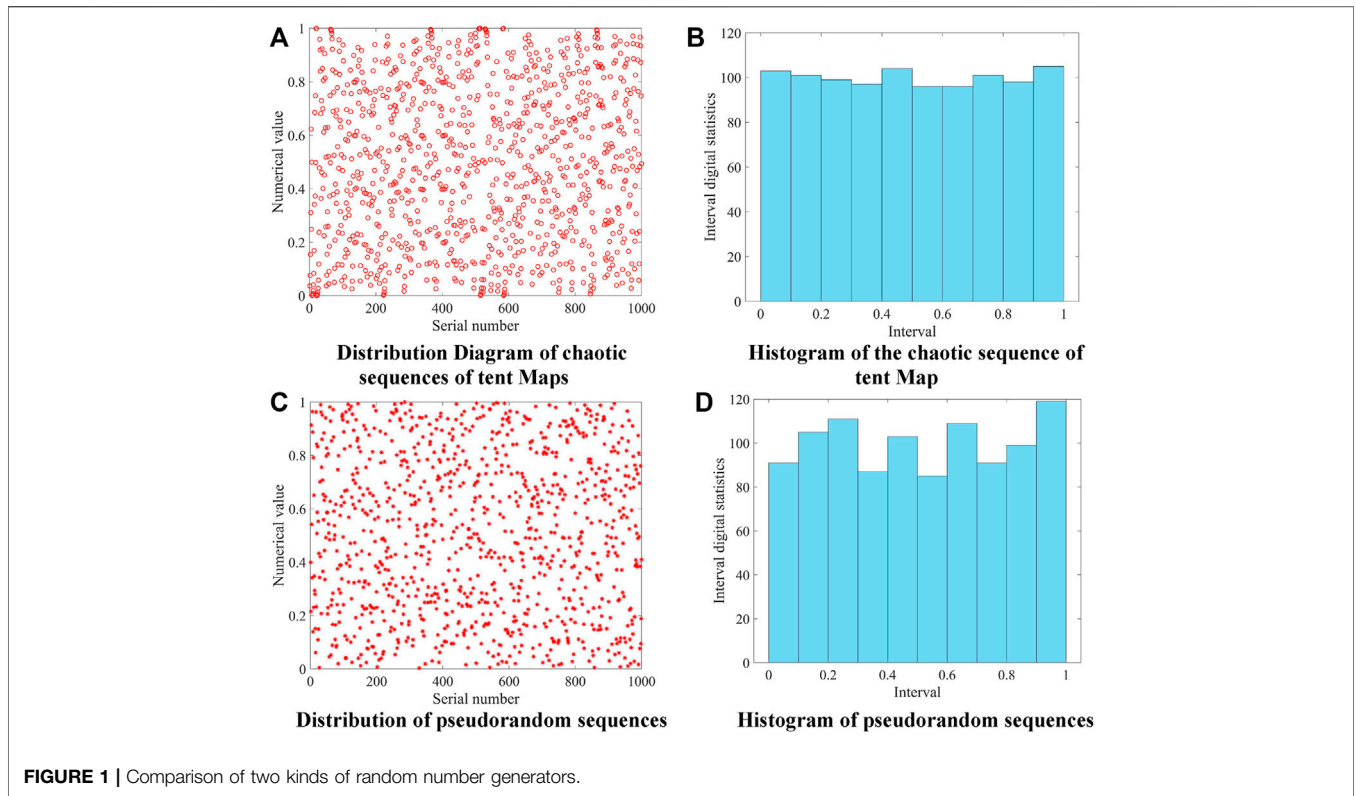
$$x_{i,j}^{T+1} = \begin{cases} Q \exp\left(-\frac{x_{\text{worst}} - x_{i,j}^T}{i^2}\right), & i > \frac{n}{2} \\ x_p^{T+1} + |x_{i,j}^T - x_p^{T+1}| A^+ L, & i \leq \frac{n}{2} \end{cases} \tag{14}$$

where  $x_{\text{worst}}$  is the individual sparrow with the weakest current energy, n is the number of joiners,  $x_p^{T+1}$  is the sparrow with the highest energy in Equation 10; A is a vector of the same dimension as the individual sparrow, with internal elements 1 and -1,  $A^+ = A^T (AA^T)^{-1} A^{-1}$ .

The accessions are renewed in two ways, preserving the distribution of the sparrow population. For the sparrow responsible for early warning

$$x_{i,j}^{T+1} = \begin{cases} x_{\text{best}}^T + \beta |x_{i,j}^T - x_{\text{best}}^T|, & f_i > f_g \\ x_{i,j}^T + K \left( \frac{x_{i,j}^T - x_{\text{worst}}}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g \end{cases} \tag{15}$$

where  $x_{\text{best}}$  is the globally optimal individual in the current action,  $f_i$  is the fitness value of the individuals, and  $f_g, f_w$  are the optimal fitness and the worst fitness, respectively, that the

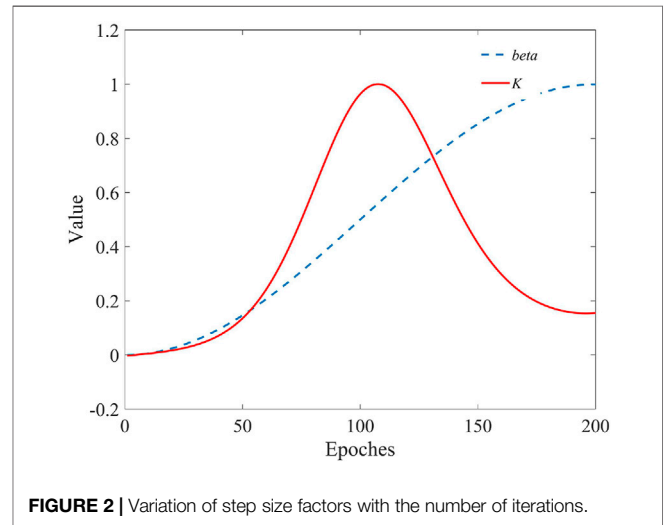


sparrow population can achieve in this action.  $\beta$ ,  $K$  are step coefficients.  $\varepsilon$  is a minimal value that prevents the denominator from being zero.

The sparrow responsible for the warning is at the edge of the population when  $f_i > f_g$  and should move closer to the position where the globally optimal sparrow is located. The sparrow responsible for sensing danger chooses to update its position towards the centre of the group in order to reduce the risk of predation when  $f_i = f_g$ . This setup aims to avoid too many individuals reaching a local optimum, and the algorithm stops iterating and falling into a local optimum.

### Initial Population Optimization Based on Tent Chaotic Mapping

In the original algorithm, the authors compared the performance of SSA and GSA, PSO, GWO on nineteen tested functions. SSA has a robust global search capability and yields high accuracy, outperforming other algorithms in multi-modal search problems. The distribution of the initial population is closely related to the performance of the algorithm. A uniformly distributed initial population can enrich the diversity of the population and thus improve the search efficiency of the algorithm. However, this consideration was not made in the original SSA, and the implementation of the algorithm relied on pseudo-random numbers to complete the initial population take. Tent chaotic sequences are stochastic and ensure the diversity of populations (Liu et al., 2016). Tent mapping can produce a uniformly



distributed initial population, which effectively prevents individuals from approaching the local optimum too early. The expression for the Tent chaotic mapping is

$$x_{i+1} = \begin{cases} 2x_i, & 0 \leq x_i \leq 0.5 \\ 2(1 - x_i), & 0.5 < x_i \leq 1 \end{cases} \quad (16)$$

In order to avoid the sequence of Tent chaotic mappings falling into small and unstable periodic points during iteration, a random variable  $\delta/N$  is introduced into the original Tent mapping

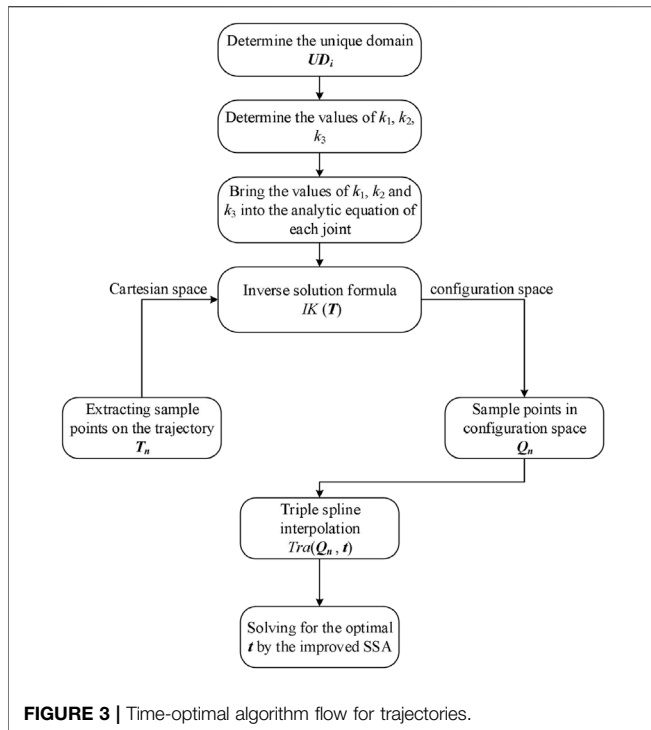


FIGURE 3 | Time-optimal algorithm flow for trajectories.

$$x_{i+1} = \begin{cases} \frac{2x_i + \delta/N - x_{\min}}{x_{\max} - x_{\min}}, & 0 \leq x_i \leq 0.5 \\ \frac{2(1 - x_i) + \delta/N - x_{\min}}{x_{\max} - x_{\min}}, & 0.5 < x_i \leq 1 \end{cases} \quad (17)$$

where the random variable  $\delta \in [0,1]$ ,  $N$  is the number of elements in the individual sparrow, and  $x_{\min}$ ,  $x_{\max}$  are the smallest and largest elements in the individual sparrow.

To further compare pseudo-random numbers and Tent chaotic sequences, 1,000 sets of random numbers with values in  $[0,1]$  were generated by two methods. The intervals were evenly

divided into 10 parts, and the frequency of occurrence of the numbers in each interval was counted. Figures 1A,C visualize the location of the random numbers generated in the two ways, while Figures 1B,D count the number of occurrences of the random numbers in each interval. It can be seen that the improved Tent mapping produces a more uniform distribution of chaotic sequences of random numbers.

### Adaptive Step Factor

The function of Eq. 15 is to avoid that too many individuals reach the optimum, and the algorithm stops iterating, thus falling into a local optimum. However, its effect is weakened because the step factor in Eq. 15 is random. The sparrows should approach the individual with the best position more quickly when  $f_i > f_g$ , thus achieving accelerated convergence. The cosine annealing algorithm is widely used in deep learning and has excellent conditioning effects. The cosine annealing algorithm was used to adjust the step size factor  $\beta$ . In early searches, sparrow populations were more widely distributed, global search capacity was more robust and local search capacity needed to be increased. Using Eq. 19, the value of  $K$  is adjusted to increase in the early period and decrease rapidly in the later period. Such an adaptation ensures that the algorithm maintains a robust global search capability in the early stages, and accelerates convergence in the later stages. The variation of the two step factors with the number of iterations is shown in Figure 2. Equations 18 and 19 are defined as adaptive step factors, which we refer to uniformly as ADF in the rest of the section

$$\beta = \beta_{\min} + 0.5(\beta_{\max} - \beta_{\min})(1 - \cos(\text{iter}_{\text{cur}}/\text{iter}_{\text{max}})) \quad (18)$$

$$K = (\text{iter}_{\text{cur}}/\text{iter}_{\text{max}}) \exp\left(\sin\left(\frac{\text{iter}_{\text{cur}}}{\text{iter}_{\text{max}}}\pi\right) - K_{\min}\right) / K_{\max} \quad (19)$$

where  $\beta_{\min}$  and  $\beta_{\max}$  are the minimum and maximum values of  $\beta$ .  $K_{\min}$  and  $K_{\max}$  are the lower and upper limits of the value of  $K$ .  $\text{iter}_{\text{cur}}$  and  $\text{iter}_{\text{max}}$  are the current number of iterations and the total number of iterations respectively.

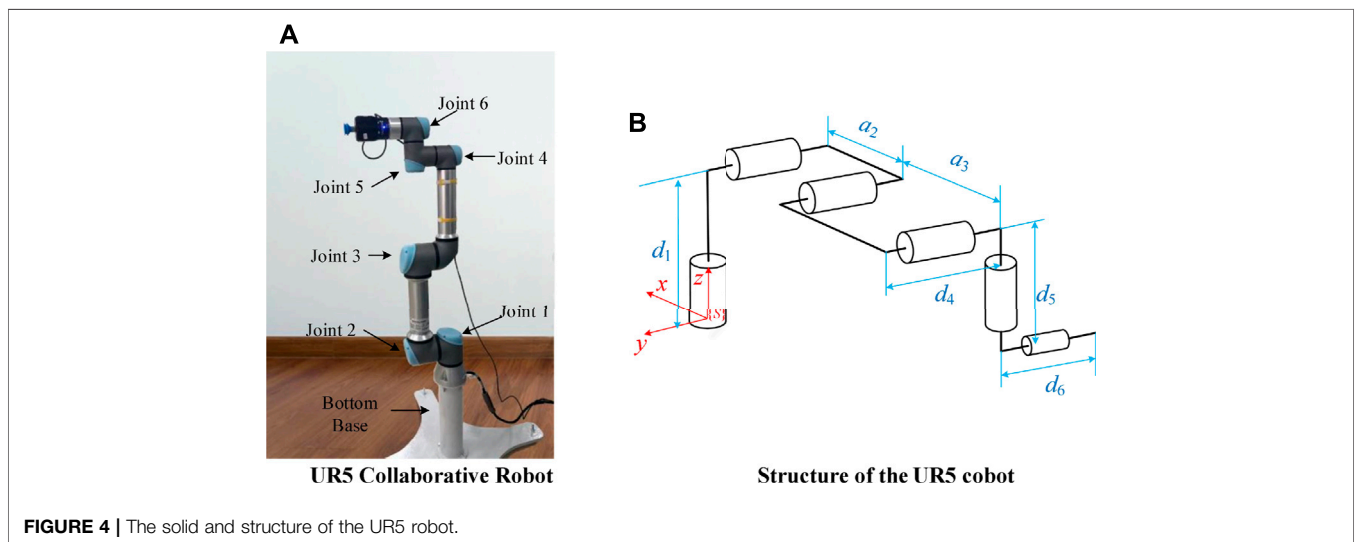


FIGURE 4 | The solid and structure of the UR5 robot.

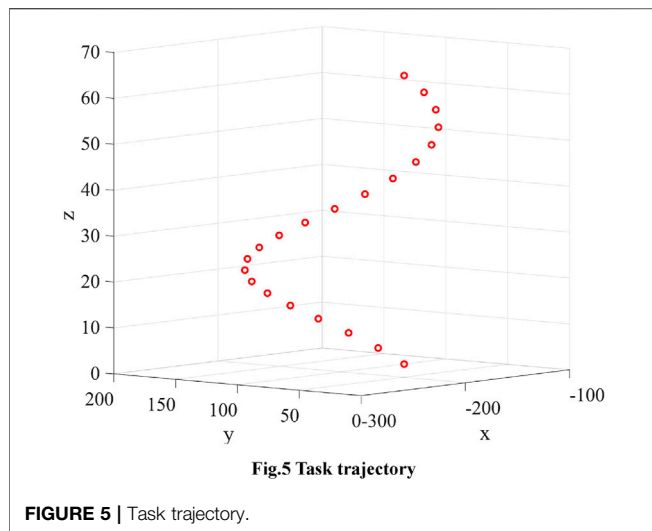


FIGURE 5 | Task trajectory.

Equations 13–15, derived from the original work of SSA (Xue and Shen, 2020), specify how the sparrow is updated and construct the basic flow of the algorithm. Based on the idealization and feasibility of the above model, the basic steps of the improved SSA can be summarized in the pseudo-code shown in **Algorithm 1**. The algorithm flow for solving the time-optimal trajectory can be obtained from the above work, as shown in **Figure 3**.  $UD_i$  in the figure is a shorthand for the unique domain,  $i = 1, 2, 3, \dots, 8$ .

**Algorithm 1.** The framework of the improved SSA

**Input:**

$G$ : the maximum iterations  
 $PD$ : the number of producers  
 $SD$ : the number of sparrows who perceive the danger  
 $R_2$ : the alarm value  
 $n$ : the number of sparrows  
 Initialize  $n$  sparrows using Eq.(17) and define the relevant parameters

**Output:**  $X_{best}, f_g$

```

1: while  $t < G$ 
2: Ranking fitness values and identifying the current best individuals and the current worst individuals.
3:  $R_2 = rand(1)$ ;
4: for  $i = 1:PD$ 
5:   Using Eq.(13) update the sparrow's location;
6: end for
7: for  $i = (PD+1):n$ 
8:   Using Eq.(14) update the sparrow's location;
9: end for
10: for  $i = 1:SD$ 
11:   Using Eqs.(15), (18) and (19) update the sparrow's location;
12: end for
13: Get the current new location;
14: If the new location is better than before, update it;
15:  $t = t + 1$ 
16: end while
17: return  $X_{best}, f_g$ 

```

TABLE 2 | Positions of every sample points.

No	x/mm	y/mm	z/mm
1	-140.000	100.000	0.000
2	-142.937	118.541	3.142
3	-151.459	135.267	6.283
4	-164.733	148.541	9.425
5	-181.459	157.063	12.566
...	...	...	...
19	-151.459	64.733	56.549
20	-142.937	81.459	59.690

TABLE 3 | The corresponding angle of the sampling point.

No	$\theta_1/\text{rad}$	$\theta_2/\text{rad}$	$\theta_3/\text{rad}$	$\theta_4/\text{rad}$	$\theta_5/\text{rad}$	$\theta_6/\text{rad}$
1	-1.5066	2.2871	-2.4364	-2.9923	1.5066	3.1416
2	-1.4931	2.2125	-2.4216	-2.9325	1.4931	3.1416
3	-1.4451	2.1280	-2.3985	-2.8711	1.4451	3.1416
4	-1.3785	2.0459	-2.3707	-2.8168	1.3785	3.1416
5	-1.3023	1.9726	-2.3418	-2.7724	1.3023	3.1416
...	...	...	...	...	...	...
19	-1.34683	2.2836	-2.5835	-2.8417	1.3468	3.1416
20	-1.46255	2.2808	-2.5912	-2.8312	1.4626	3.1416

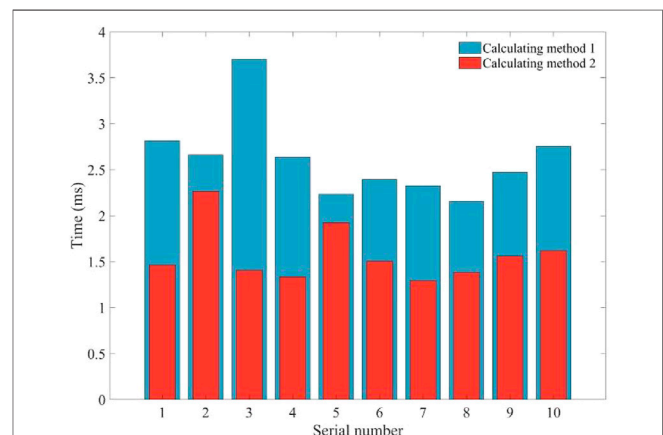


FIGURE 6 | Time results for ten calculations of the two inverse solution schemes.

## EXPERIMENTS BASED ON UR5 COBOT

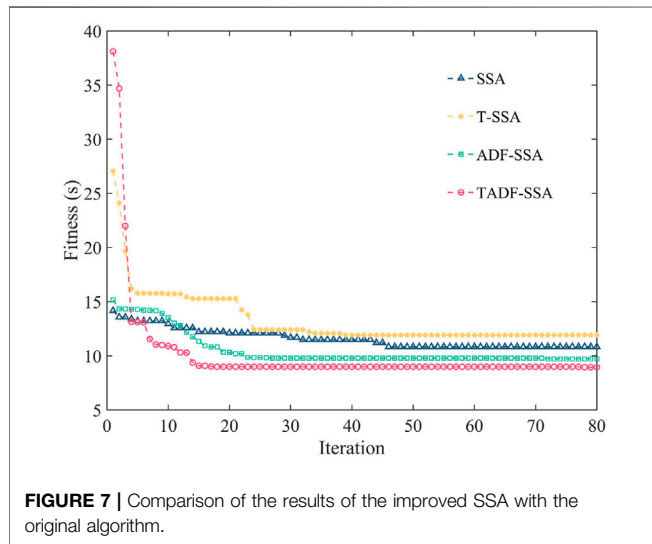
Experiments were carried out on the UR5 cobot platform in order to verify the validity of the work presented earlier. The solid and structure of the UR5 robot are shown in **Figures 4A,B**, and the DH parameters are shown in **Table 1**.

The task trajectory is a spiral line with a circle centre position of (-200,100)mm, a radius of 60 mm and a pitch of 10 mm in a Cartesian coordinate system. The trajectory of the helix in the Cartesian system is shown in **Figure 5** and the coordinates of the sampling points are shown in **Table 2**.



**TABLE 4** | Constraint conditions of each joint.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
$V_{max}$ (rad/s)	1.7453	1.6581	1.7453	2.6180	2.2689
$A_{max}$ (rad/s <sup>2</sup> )	0.7854	0.6981	1.3090	1.2217	1.5708
$J_{max}$ (rad/s <sup>3</sup> )	1.0472	1.0472	0.9599	1.2217	1.3090



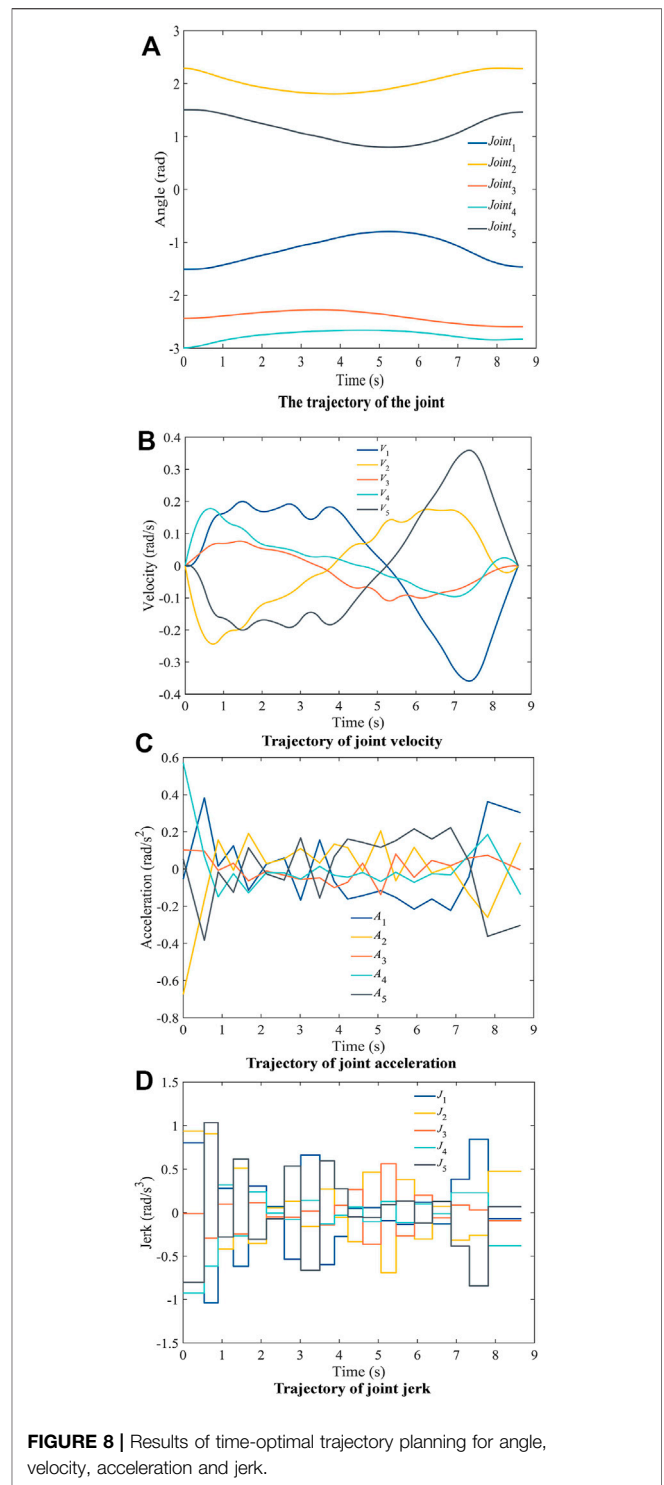
**FIGURE 7** | Comparison of the results of the improved SSA with the original algorithm.

**TABLE 5** | Average of convergence after 10 runs of the four algorithms at the terminal of Intel(R) i7-9750H CUP@ 2.60 GHz.

	SSA	T-SSA	ADF-SSA	TADF-SSA
Iterations at convergence	47.8	34.3	24.3	18.2
Fitness at convergence (s)	10.87	12.14	9.91	9.04

The efficiency of the inverse solution calculation during robot motion is important and affects the efficiency of the overall motion. A single calculation can derive the joint angles for all sample points by Eq. 4 in Section 3. The joint angles corresponding to each sample point are obtained by inverse kinematics as shown in Table 3. Since the pose did not change during the motion along the trajectory, no motion occurred for joint 6, no trajectory planning was required for joint 6.

Compared to the method of calculating eight sets of inverse solutions and then filtering them by the principle of minimum displacement (known as calculation method 1), the method of joint space division (known as calculation method 2) is more time-efficient. The two inverse solution calculation methods were run ten times in Matlab 2019b, and the calculation time results are shown in Figure 6. The average time for ten runs of the two methods was 2.6149 and 1.5752 ms respectively, with the joint space division based method able to save 39.76% of



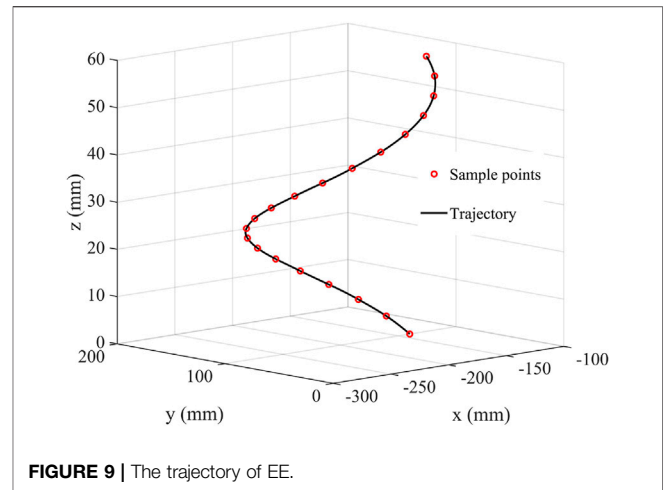
**FIGURE 8** | Results of time-optimal trajectory planning for angle, velocity, acceleration and jerk.

the time. This proportion increases linearly with the number of sampling points.

The maximum velocity and maximum acceleration constraints for each joint are shown in Table 4. We did controlled experiments in the same environment to further investigate the advantages of the two improvements to SSA in this paper. The maximum number of

**TABLE 6** | Interpolation time for each segment of the trajectory.

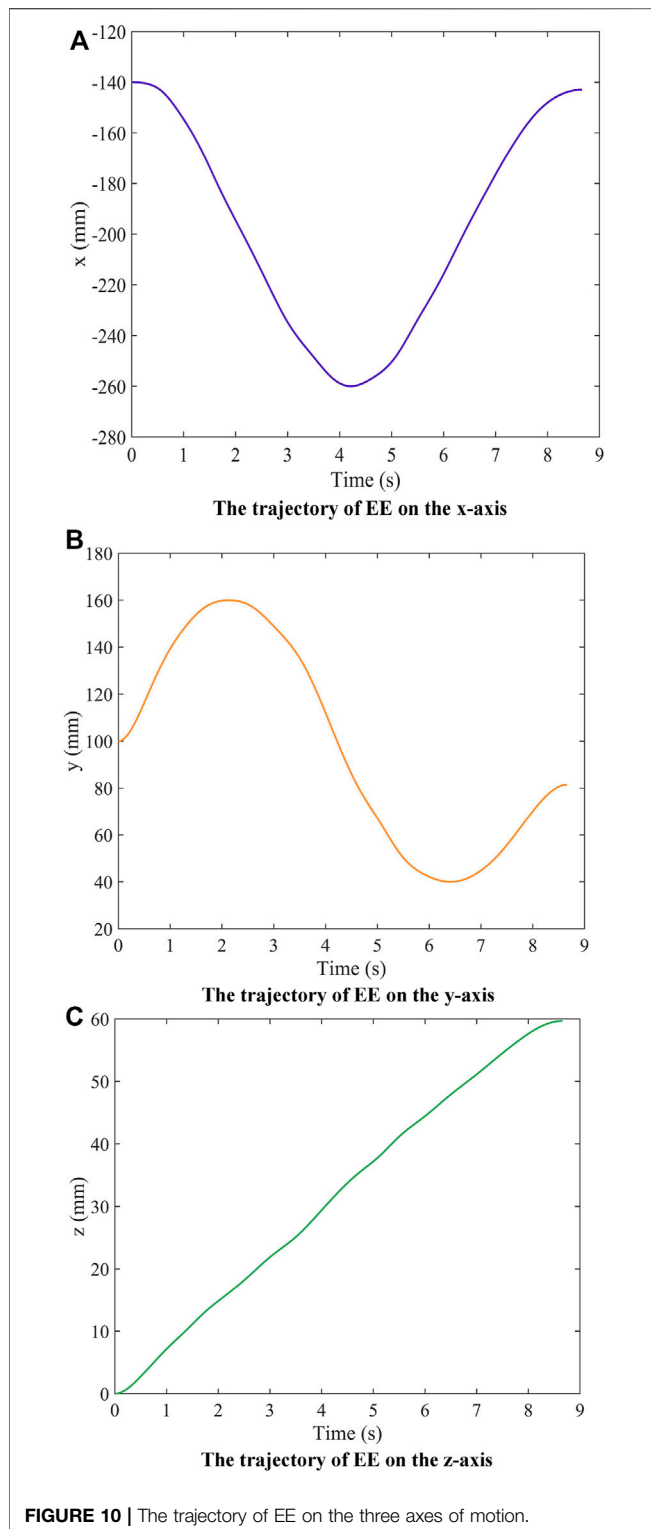
	$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$	$h_9$	$h_{10}$	$h_{11}$	$h_{12}$	$h_{13}$	$h_{14}$	$h_{15}$	$h_{16}$	$h_{17}$	$h_{18}$	$h_{19}$
Time/s	0.5439	0.3541	0.3914	0.3878	0.4609	0.4540	0.4237	0.4882	0.3734	0.3460	0.3829	0.4649	0.3870	0.4704	0.4555	0.4819	0.4686	0.4803	0.8469



iterations is 80, the population size is 100, and the number of discoverers is 20% of the entire population. The sparrows responsible for scouting were 10% of the population, and  $ST = 0.5$  in Eq. 13. To compare the effects of the two improvements on SSA, the same trajectory is optimized with the same initial parameters. SSA combined with Tent chaotic mapping is referred to as T-SSA. SSA combined with an adaptive step factor is referred to as ADF-SSA. The SSA with both improvements is called TADF-SSA. To verify the stability of the algorithms, the variations of the fitness functions of the four algorithms are run ten times separately at the same hardware level. The optimal performance of each algorithm is shown in Figure 7. The results of the ten runs are averaged and recorded in Table 5.

From Figure 7, we can see that the best convergence result for SSA is 10.83s after 46 iterations. T-SSA converges to a minimum value of 12.07s after 34 iterations. ADF-SSA converges to a minimum value of 9.85s after 23 iterations. TADF-SSA converges to a minimum value of 8.99s after 18 iterations. As can be seen from Table 5, the average results of the ten runs float around the best results, and the algorithm's performance is stable. The results of T-SSA and SSA shows that using Tent chaotic mapping alone does not improve the algorithm significantly. This is since the Tent chaotic mapping reduces the effect of population precocity, while the search capability of the SSA algorithm itself causes this result. The results of ADF-SSA and SSA shows that the adaptive step factor (ADF) can improve the global search capability and efficiency of the algorithm. TADF-SSA performed best in all results, which demonstrates the effectiveness of the improvements to the SSA algorithm in this paper.

The initial time was set to 50s, and the trajectory obtained after time optimization by the improved SSA is shown in Figure 8. The movement time of the whole trajectory is 8.6618s (the average of 5 runs is taken as the final time to ensure the algorithm's stability), which is 82.68% shorter. The interpolation time for each segment is shown in Table 6. The trajectory of each joint's angle (as shown in Figure 8A) and velocity (as shown in Figure 8B) changes is very smooth, and the maximum value is less than the constraint. The acceleration profile is continuous, as shown in Figure 8C, and the maximum value of the angular acceleration of each joint is less than the constraint. The jerk of each joint angle is also much smaller than



the constraint, as shown in **Figure 8D**, which ensures that the motor is protected from large shocks.

The trajectory of the obtained joint angles is mapped by forward kinematics to the trajectory of EE in a Cartesian coordinate system as

shown in **Figure 9**. It can be seen that the trajectory of EE in Cartesian space is very smooth and passes through each sampling point. The result is consistent with engineering applications and practical requirements.

The variation of the components of the end trajectory mapped to the three axes in Cartesian space is shown in **Figures 10A-C** (correspond to the trajectories of axes x, y, and z, respectively). The trajectories are smooth and continuous in all three coordinates, which indicates that the EE is moving smooth and stable in the set trajectory.

## CONCLUSION

An optimal time trajectory planning method is proposed in terms of both inverse kinematic solution and time optimization. Trajectory planning is accomplished in the joint space through cubic spline interpolation, and the joint space is refined to improve the inverse solution calculation time. The sparrow search algorithm is optimized for the initial population generation method and step factor update. An improved tent chaotic mapping improves the rationality of the initial population distribution. The global search capability of the algorithm is improved by combining the step size factor of the cosine annealing algorithm. A time-optimal trajectory was obtained by an improved sparrow search algorithm. Simulation experiments were carried out on the UR5 collaborative robot, and the results showed that the time of the obtained trajectory was considerably optimized while satisfying the constraints. The feasibility and effectiveness of the algorithm were verified.

The method proposed in this paper for calculating time-optimal trajectories is applicable to tandem robots of any configuration. The method of calculating the inverse solution when planning trajectories can improve the efficiency of robot motion, but this method relies on the robot's structure and has not been generalized to robots of general configuration, which requires further theoretical study.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

## AUTHOR CONTRIBUTIONS

XZ provided the idea and wrote the article, FX derives the formula for the inverse kinematics in the article. XT was responsible for checking and touching up the language of the article, and YS and BC provided all of the research funding.

## FUNDING

This work was supported by grants of the National Natural Science Foundation of China (Grant Nos. 52075530, 51575407, 51975324,

51505349, 61733011, 41906177); the Grants of Hubei Provincial Department of Education (D20191105); the Grants of National Defense PreResearch Foundation of Wuhan University of Science and Technology (GF201705) and Open Fund of the Key Laboratory for Metallurgical Equipment and Control of

Ministry of Education in Wuhan University of Science and Technology (2018B07, 2019B13) and Open Fund of Hubei Key Laboratory of Hydroelectric Machinery Design & Maintenance in China Three Gorges University (2020KJX02, 2021KJX13).

## REFERENCES

- Abdollahzadeh, B., Gharehchopogh, F. S., and Mirjalili, S. (2021b). African Vultures Optimization Algorithm: A New Nature-Inspired Metaheuristic Algorithm for Global Optimization Problems. *Comput. Ind. Eng.* 158, 107408. doi:10.1016/j.cie.2021.107408
- Abdollahzadeh, B., Soleimani Gharehchopogh, F., and Mirjalili, S. (2021a). Artificial gorilla Troops Optimizer: A New Nature-inspired Metaheuristic Algorithm for Global Optimization Problems. *Int. J. Intell. Syst.* 36 (10), 5887–5958. doi:10.1002/int.22535
- Abedi, M., and Gharehchopogh, F. S. (2020). An Improved Opposition Based Learning Firefly Algorithm with Dragonfly Algorithm for Solving Continuous Optimization Problems. *Ida* 24 (2), 309–338. doi:10.3233/IDA-194485
- Bai, D., Sun, Y., Tao, B., Tong, X., Xu, M., Jiang, G., et al. (2022). Improved Single Shot Multibox Detector Target Detection Method Based on Deep Feature Fusion. *Concurrency Comput.* 34 (4), e6614. doi:10.1002/cpe.6614
- Benyamin, A., Farhad, S. G., and Saeid, B. (2021). Discrete farmland Fertility Optimization Algorithm with metropolis Acceptance Criterion for Traveling Salesman Problems. *Int. J. Intell. Syst.* 36 (3), 1270–1303. doi:10.1002/int.22342
- Biagiotti, L., and Melchiorri, C. (2008). *Trajectory Planning for Automatic Machines and Robots*. Springer Science & Business Media. doi:10.1007/978-3-540-85629-0
- Bobrow, J. E., Dubowsky, S., and Gibson, J. S. (1985). Time-optimal Control of Robotic Manipulators along Specified Paths. *Int. J. robotics Res.* 4 (3), 3–17. doi:10.1177/027836498500400301
- Chen, M. W., and Zalzal, A. M. S. (1997). Dynamic Modelling and Genetic-Based Trajectory Generation for Non-holonomic mobile Manipulators. *Control. Eng. Pract.* 5 (1), 39–48. doi:10.1016/S0967-0661(96)00205-5
- Chen, T., Peng, L., Yang, J., Cong, G., and Li, G. (2021a). Evolutionary Game of Multi-Subjects in Live Streaming and Governance Strategies Based on Social Preference Theory during the COVID-19 Pandemic. *Mathematics* 9 (21), 2743. doi:10.3390/math9212743
- Chen, T., Qiu, Y., Wang, B., and Yang, J. (2022). Analysis of Effects on the Dual Circulation Promotion Policy for Cross-Border E-Commerce B2B Export Trade Based on System Dynamics during COVID-19. *Systems* 10 (1), 13. doi:10.3390/systems10010013
- Chen, T., Yin, X., Yang, J., Cong, G., and Li, G. (2021b). Modeling Multi-Dimensional Public Opinion Process Based on Complex Network Dynamics Model in the Context of Derived Topics. *Axioms* 10 (4), 270. doi:10.3390/axioms10040270
- Cheng, Y., Li, G., Li, J., Sun, Y., Jiang, G., Zeng, F., et al. (2020). Visualization of Activated Muscle Area Based on sEMG. *Ijs* 38 (3), 2623–2634. doi:10.3233/JIFS-179549
- Cheng, Y., Li, G., Yu, M., Jiang, D., Yun, J., Liu, Y., et al. (2021). Gesture Recognition Based on Surface Electromyography -feature Image. *Concurrency Computat Pract. Exper* 33 (6), e6051. doi:10.1002/cpe.6051
- Constantinescu, D., and Croft, E. A. (2000). Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators along Specified Paths. *J. Robotic Syst.* 17 (5), 233–249. doi:10.1002/(sici)1097-4563(200005)17:5<233::aid-rob1>3.0.co;2-y
- Constantinescu, D. (1998). *Smooth Time Optimal Trajectory Planning for Industrial Manipulators*. Doctoral dissertation, University of British Columbia.
- Duan, H., Sun, Y., Jiang, D., and Cheng, W. (2021). Gesture Recognition Based on Multi-modal Feature Weight. *Concurr Comp-Pract E.* 33 (5), e5991. doi:10.1002/cpe.5991
- Gasparetto, A., Boscaroli, P., Lanzutti, A., and Vidoni, R. (2012). Trajectory Planning in Robotics. *Math.Comput.Sci.* 6 (3), 269–279. doi:10.1007/s11786-012-0123-8
- Ghafori, S., and Gharehchopogh, F. S. (2012). Advances in Spotted Hyena Optimizer: a Comprehensive Survey. *Arch. Computat Methods Eng.* 1, 1–22. doi:10.1007/s11831-021-09624-4
- Gharehchopogh, F. S., and Abdollahzadeh, B. (2021). An Efficient Harris Hawk Optimization Algorithm for Solving the Travelling Salesman Problem. *Cluster Comput.* 1, 1–25. doi:10.1007/s10586-021-03304-5
- Gharehchopogh, F. S. (2022). Advances in Tree Seed Algorithm: A Comprehensive Survey. *Arch. Computat Methods Eng.* 1, 1–24. doi:10.1007/s11831-021-09698-0
- Gharehchopogh, F. S., Farnad, B., and Alizadeh, A. (2021a). A Modified farmland Fertility Algorithm for Solving Constrained Engineering Problems. *Concurrency Computat Pract. Exper* 33 (17), e6310. doi:10.1002/cpe.6310
- Gharehchopogh, F. S., Maleki, L., and Dizaji, Z. A. (2021b). Chaotic Vortex Search Algorithm: Metaheuristic Algorithm for Feature Selection. *Evol. Intel.* 1, 1–32. doi:10.1007/s12065-021-00590-1
- Goldanloo, M. J., and Gharehchopogh, F. S. (2021). A Hybrid OBL-Based Firefly Algorithm with Symbiotic Organisms Search Algorithm for Solving Continuous Optimization Problems. *J. Supercomput* 78, 3998–4031. doi:10.1007/s11227-021-04015-9
- Hao, Z., Wang, Z., Bai, D., and Zhou, S. (2021). Towards the Steel Plate Defect Detection: Multidimensional Feature Information Extraction and Fusion. *Concurr Comp-Pract E.* 33 (21), e6384. doi:10.1002/CPE.6384
- Hao, Z., Wang, Z., Bai, D., Tao, B., Tong, X., and Chen, B. (2022). Intelligent Detection of Steel Defects Based on Improved Split Attention Networks. *Front. Bioeng. Biotechnol.* 9, 810876. doi:10.3389/fbioe.2021.810876
- He, Y., Li, G., Liao, Y., Sun, Y., Kong, J., Jiang, G., et al. (2019). Gesture Recognition Based on an Improved Local Sparse Representation Classification Algorithm. *Cluster Comput.* 22 (5), 10935–10946. doi:10.1007/s10586-017-1237-1
- Hu, J., Sun, Y., Li, G., Jiang, G., and Tao, B. (2019). Probability Analysis for Grasp Planning Facing the Field of Medical Robotics. *Measurement* 141, 227–234. doi:10.1016/j.measurement.2019.03.010
- Huang, L., Fu, Q., He, M., Jiang, D., and Hao, Z. (2021). Detection Algorithm of Safety Helmet Wearing Based on Deep Learning. *Concurr Comp-Pract E.* 33 (13), e6234. doi:10.1002/cpe.6234
- Huang, L., Fu, Q., Li, G., Luo, B., Chen, D., and Yu, H. (2019). Improvement of Maximum Variance Weight Partitioning Particle Filter in Urban Computing and Intelligence. *IEEE Access* 7, 106527–106535. doi:10.1109/access.2019.2932144
- Huang, L., He, M., Tan, C., Jiang, D., Li, G., and Yu, H. (2020). Jointly Network Image Processing: Multi-task Image Semantic Segmentation of Indoor Scene Based on CNN. *IET image process* 14 (15), 3689–3697. doi:10.1049/iet-ipr.2020.0088
- Huang, P., Liu, G., Yuan, J., and Xu, Y. (2008). “Multi-objective Optimal Trajectory Planning of Space Robot Using Particle Swarm Optimization,” in *International Symposium on Neural Networks* (Berlin, Heidelberg: Springer), 171–179. doi:10.1007/978-3-540-87734-9-20
- Huang, P., and Xu, Y. (2006). “PSO-based Time-Optimal Trajectory Planning for Space Robot with Dynamic Constraints,” in *IEEE International Conference on Robotics and Biomimetics* (Kunming, China: IEEE), 1402–1407. doi:10.1109/ROBIO.2006.340134
- Jiang, D., Li, G., Sun, Y., Hu, J., Yun, J., and Liu, Y. (2021). Manipulator Grabbing Position Detection with Information Fusion of Color Image and Depth Image Using Deep Learning. *J. Ambient Intell. Hum. Comput* 12 (12), 10809–10822. doi:10.1007/s12652-020-02843-w
- Jiang, D., Li, G., Sun, Y., Kong, J., Tao, B., and Chen, D. (2019). Grip Strength Forecast and Rehabilitative Guidance Based on Adaptive Neural Fuzzy Inference System Using sEMG. *Pers Ubiquit Comput.* 1, 1–10. doi:10.1007/s00779-019-01268-3
- Jiang, D., Li, G., Sun, Y., Kong, J., and Tao, B. (2019). Gesture Recognition Based on Skeletonization Algorithm and CNN with ASL Database. *Multimed Tools Appl.* 78 (21), 29953–29970. doi:10.1007/s11042-018-6748-0
- Jiang, D., Li, G., Tan, C., Huang, L., Sun, Y., and Kong, J. (2021). Semantic Segmentation for Multiscale Target Based on Object Recognition Using the Improved Faster-RCNN Model. *Future Generation Computer Syst.* 123, 94–104. doi:10.1016/j.future.2021.04.019

- Jiang, D., Zheng, Z., Li, G., Sun, Y., Kong, J., Jiang, G., et al. (2019). Gesture Recognition Based on Binocular Vision. *Cluster Comput.* 22 (6), 13261–13271. doi:10.1007/s10586-018-1844-5
- Kang Shin, K., and McKay, N. (1985). Minimum-time Control of Robotic Manipulators with Geometric Path Constraints. *IEEE Trans. Automat. Contr.* 30 (6), 531–541. doi:10.1109/tac.1985.1104009
- Kant, K., and Zucker, S. W. (1986). Toward Efficient Trajectory Planning: The Path-Velocity Decomposition. *Int. J. robotics Res.* 5 (3), 72–89. doi:10.1177/027836498600500304
- Li, B., Sun, Y., Li, G., Kong, J., Jiang, G., Jiang, D., et al. (2019/2019). Gesture Recognition Based on Modified Adaptive Orthogonal Matching Pursuit Algorithm. *Cluster Comput.* 22 (1), 503–512. doi:10.1007/s10586-017-1231-7
- Li, C., Li, G., Jiang, G., Chen, D., and Liu, H. (2020). Surface EMG Data Aggregation Processing for Intelligent Prosthetic Action Recognition. *Neural Comput. Applic* 32 (22), 16795–16806. doi:10.1007/s00521-018-3909-z
- Li, G., Jiang, D., Zhou, Y., Jiang, G., Kong, J., and Manogaran, G. (2019b). Human Lesion Detection Method Based on Image Information and Brain Signal. *IEEE Access* 7, 11533–11542. doi:10.1109/ACCESS.2019.2891749
- Li, G., Li, J., Ju, Z., Sun, Y., and Kong, J. (2019a). A Novel Feature Extraction Method for Machine Learning Based on Surface Electromyography from Healthy Brain. *Neural Comput. Applic* 31 (12), 9013–9022. doi:10.1007/s00521-019-04147-3
- Li, G., Zhang, L., Sun, Y., and Kong, J. (2019c). Towards the SEMG Hand: Internet of Things Sensors and Haptic Feedback Application. *Mul.Timed Tools Appl.* 78 (21), 29765–29782. doi:10.1007/s11042-018-6293-x
- Li, Z., Li, G., Jiang, G., Fang, Y., Ju, Z., and Liu, H. (2015). Computation of Grasping and Manipulation for Multi-Fingered Robotic Hands. *J Comput. Theor. Nanosci* 12 (3), 6192–6197. doi:10.1166/jctn.2015.4655
- Liao, S., Li, G., Li, J., Jiang, D., Jiang, G., Sun, Y., et al. (2020). Multi-object Intergroup Gesture Recognition Combined with Fusion Feature and KNN Algorithm. *Ijs* 38 (3), 2725–2735. doi:10.3233/JIFS-179558
- Liao, S., Li, G., Wu, H., Jiang, D., Liu, Y., Yun, L., et al. (2021). Occlusion Gesture Recognition Based on Improved SSD. *Concurr Comp-Pract E.* 33 (6), e6063. doi:10.1002/cpe.6063
- Liu, B., and Rodriguez, D. (2021). Renewable Energy Systems Optimization by a New Multi-Objective Optimization Technique: A Residential Building. *J. Building Eng.* 35, 102094. doi:10.1016/j.jobe.2020.102094
- Liu, G., Shu, C., Liang, Z., Peng, B., and Cheng, L. (2021). A Modified Sparrow Search Algorithm with Application in 3d Route Planning for UAV. *Sensors* 21 (4), 1224. doi:10.3390/s21041224
- Liu, H., Lai, X., and Wu, W. (2013). Time-optimal and Jerk-Continuous Trajectory Planning for Robot Manipulators with Kinematic Constraints. *Robotics and Computer-Integrated Manufacturing* 29 (2), 309–317. doi:10.1016/j.rcim.2012.08.002
- Liu, L., Sun, S. Z., Yu, H., Yue, X., and Zhang, D. (2016). A Modified Fuzzy C-Means (FCM) Clustering Algorithm and its Application on Carbonate Fluid Identification. *J. Appl. Geophys.* 129, 28–35. doi:10.1016/j.jappgeo.2016.03.027
- Liu, X., Jiang, D., Tao, B., Jiang, G., Sun, Y., Kong, J., et al. (2022). Genetic Algorithm-Based Trajectory Optimization for Digital Twin Robots. *Front. Bioeng. Biotechnol.* 9, 793782. doi:10.3389/fbioe.2021.793782
- Liu, Y., Jiang, D., Tao, B., Qi, J., Jiang, G., Yun, J., et al. (2022). Grasping Posture of Humanoid Manipulator Based on Target Shape Analysis and Force Closure. *Alexandria Eng. J.* 61 (5), 3959–3969. doi:10.1016/j.aej.2021.09.017
- Liu, Y., Jiang, D., Yun, J., Sun, Y., Li, C., Jiang, G., et al. (2021). Self-tuning Control of Manipulator Positioning Based on Fuzzy PID and PSO Algorithm. *Front. Bioeng. Biotechnol.* 9, 1. doi:10.3389/fbioe.2021.817723
- Liu, Y., Li, C., Jiang, D., Chen, B., Sun, N., Cao, Y., et al. (2022a). Wrist Angle Prediction under Different Loads Based on GA-ELM Neural Network and Surface Electromyography. *Concurrency Comput.* 34 (3), e6574. doi:10.1002/cpe.6574
- Liu, Y., Xiao, F., Tong, X., Tao, B., Xu, M., Jiang, G., Chen, B., Cao, Y., and Sun, N. (2022b). Manipulator Trajectory Planning Based on Work Subspace Division. *Concurrency Comput.* 34 (5), e6710. doi:10.1002/cpe.6710
- Liu, Y., Xu, M., Jiang, G., Tong, X., Yun, J., Liu, Y., et al. (2022c). Target Localization in Local Dense Mapping Using RGBD SLAM and Object Detection. *Concurr Comp-Pract E.* 34 (4), e6655. doi:10.1002/cpe.6655
- Luo, B., Sun, Y., Li, G., Chen, D., and Ju, Z. (2020). Decomposition Algorithm for Depth Image of Human Health Posture Based on Brain Health. *Neural Comput. Applic* 32 (10), 6327–6342. doi:10.1007/s00521-019-04141-9
- Ma, R., Zhang, L., Li, G., Jiang, D., Xu, S., and Chen, D. (2020). Grasping Force Prediction Based on sEMG Signals. *Alexandria Eng. J.* 59 (3), 1135–1147. doi:10.1016/j.aej.2020.01.007
- Mohammadzadeh, H., and Gharehchopogh, F. S. (2021). A Multi-agent System Based for Solving High-dimensional Optimization Problems: A Case Study on Email Spam Detection. *Int. J. Commun. Syst.* 34 (3), e4670. doi:10.1002/dac.4670
- Park, F. C. (1995). Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design. *J.Mech.Design.* 117 (1), 48–54. doi:10.1115/1.2826116
- Shiller, Z. (1996). Time-energy Optimal Control of Articulated Systems with Geometric Path Constraints. *IEEE Int. Conf. Robotics Automation* 1, 139–143. doi:10.1109/ROBOT.1994.350931
- Stilman, M. (2010). Global Manipulation Planning in Robot Joint Space with Task Constraints. *IEEE Trans. Robot.* 26 (3), 576–584. doi:10.1109/tro.2010.2044949
- Sun, Y., Hu, J., Li, G., Jiang, G., Xiong, H., Tao, B., et al. (2020a). Gear Reducer Optimal Design Based on Computer Multimedia Simulation. *J. Supercomput* 76 (6), 4132–4148. doi:10.1007/s11227-018-2255-3
- Sun, Y., Li, C., Li, G., Jiang, G., Jiang, D., Liu, H., et al. (2018). Gesture Recognition Based on Kinect and sEMG Signal Fusion. *Mobile Netw. Appl.* 23 (4), 797–805. doi:10.1007/s11036-018-1008-0
- Sun, Y., Tian, J., Jiang, D., Tao, B., Liu, Y., Yun, J., et al. (2020b). Numerical Simulation of thermal Insulation and Longevity Performance in New Lightweight Ladle. *Concurrency Computat Pract. Exper* 32 (22), e5830. doi:10.1002/cpe.5830
- Sun, Y., Weng, Y., Luo, B., Li, G., Tao, B., Jiang, D., et al. (2020c). Gesture Recognition Algorithm Based on Multi-scale Feature Fusion in RGB-D Images. *IET image process* 14 (15), 3662–3668. doi:10.1049/iet-ivr.2020.0148
- Sun, Y., Xu, C., Li, G., Xu, W., Kong, J., Jiang, D., et al. (2020d). Intelligent Human Computer Interaction Based on Non Redundant EMG Signal. *Alexandria Eng. J.* 59 (3), 1149–1157. doi:10.1016/j.aej.2020.01.015
- Sun, Y., Yang, Z., Tao, B., Jiang, G., Hao, Z., and Chen, B. (2021). Multiscale Generative Adversarial Network for Real-world Super-resolution. *Concurrency Computat Pract. Exper* 33 (21), e6430. doi:10.1002/CPE.6430
- Sun, Y., Zhao, Z., Jiang, D., Tong, X., and Tao, B. (2022). Low-illumination Image Enhancement Algorithm Based on Improved Multi-Scale Retinex and ABC Algorithm Optimization. *Front. Bioeng. Biotechnol.* 1, 1. doi:10.3389/fbioe.2022.865820
- Tan, C., Sun, Y., Li, G., Jiang, G., Chen, D., and Liu, H. (2020). Research on Gesture Recognition of Smart Data Fusion Features in the IoT. *Neural Comput. Applic* 32 (22), 16917–16929. doi:10.1007/s00521-019-04023-0
- Tao, B., Liu, Y., Huang, L., Chen, G., and Chen, B. (2022a). 3D Reconstruction Based on Photoelastic Fringes. *Concurrency Computat Pract. Exper* 34 (1), e6481. doi:10.1002/CPE.6481
- Tao, B., Wang, Y., Qian, X., and Tong, X. (2022b). Photoelastic Stress Field Recovery Using Deep Convolutional Neural Network. *Front. Bioeng. Biotechnol.* 1, 1. doi:10.3389/fbioe.2022.818112
- Thompson, S. E., and Patel, R. V. (1987). Formulation of Joint Trajectories for Industrial Robots Using B-Splines. *IEEE Trans. Ind. Electron.* IE-34 (2), 192–199. doi:10.1109/tie.1987.350954
- Tian, J., Cheng, W., Sun, Y., Li, G., Jiang, D., Jiang, G., et al. (2020). Gesture Recognition Based on Multilevel Multimodal Feature Fusion. *Ijs* 38 (3), 2539–2550. doi:10.3233/JIFS-179541
- Verscheure, D., Demeulenaere, B., Swevers, J., De Schutter, J., and Diehl, M. (2009). Time-optimal Path Tracking for Robots: A Convex Optimization Approach. *IEEE Trans. Automat. Contr.* 54 (10), 2318–2327. doi:10.1109/tac.2009.2028959
- Weng, Y., Sun, Y., Jiang, D., Tao, B., Liu, Y., Yun, J., et al. (2021). Enhancement of Real-time Grasp Detection by Cascaded Deep Convolutional Neural Networks. *Concurrency Computat Pract. Exper* 33 (5), e5976. doi:10.1002/cpe.5976
- Wenger, P. (1992). “A New General Formalism for the Kinematic Analysis of All Nonredundant Manipulators,” in *IEEE International Conference on Robotics and Automation* (Nice, France: IEEE), 442–447. doi:10.1109/ROBOT.1992.220300
- Wenger, P. (2019). “Cuspidal Robots,” in *Singular Configurations of Mechanisms and Manipulators* (Cham: Springer), 67–99. doi:10.1007/978-3-030-05219-5\_3

- Wu, X., Jiang, D., Yun, J., Liu, X., Sun, Y., Tao, B., et al. (2022). Attitude Stabilization Control of Autonomous Underwater Vehicle Based on Decoupling Algorithm and PSO-ADRC. *Front. Bioeng. Biotech.* 1, 1. doi:10.3389/fbioe.2022.843020
- Xiao, F., Li, G., Jiang, D., Xie, Y., Yun, J., Liu, Y., Huang, L., and Fang, Z. (2021). An Effective and Unified Method to Derive the Inverse Kinematics Formulas of General Six-DOF Manipulator with Simple Geometry. *Mechanism Machine Theor.* 159, 104265. doi:10.1016/j.mechmachtheory.2021.104265
- Xu, M., Zhang, Y., Wang, S., and Jiang, G. (2022). Genetic-Based Optimization of 3D Burch-Schneider Cage with Functionally Graded Lattice Material. *Front. Bioeng. Biotechnol.* 10, 819005. doi:10.3389/fbioe.2022.819005
- Xu, W., Li, C., Wang, X., Liu, Y., Liang, B., and Xu, Y. (2009). Study on Non-holonomic Cartesian Path Planning of a Free-Floating Space Robotic System. *Adv. Robotics* 23 (1-2), 113–143. doi:10.1163/156855308X392708
- Xue, J., and Shen, B. (2020). A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Syst. Sci. Control. Eng.* 8 (1), 22–34. doi:10.1080/21642583.2019.1708830
- Yang, Y., Huang, D., and Dong, X. (2019). Enhanced Neural Network Control of Lower Limb Rehabilitation Exoskeleton by Add-On Repetitive Learning. *Neurocomputing* 323, 256–264. doi:10.1016/j.neucom.2018.09.085
- Yang, Z., Jiang, D., Sun, Y., Tao, B., Tong, X., Jiang, G., Xu, M., Yun, J., Liu, Y., Chen, B., and Kong, J. (2021). Dynamic Gesture Recognition Using Surface EMG Signals Based on Multi-Stream Residual Network. *Front. Bioeng. Biotechnol.* 9, 1. doi:10.3389/fbioe.2021.779353
- Yoshikawa, T. (1985). Manipulability of Robotic Mechanisms. *Int. J. Robotics Res.* 4 (2), 3–9. doi:10.1177/027836498500400201
- Yu, M., Li, G., Jiang, D., Jiang, G., Tao, B., and Chen, D. (2019). Hand Medical Monitoring System Based on Machine Learning and Optimal EMG Feature Set. *Pers Ubiquit Comput.* 1, 1. doi:10.1007/s00779-019-01285-2
- Yu, M., Li, G., Jiang, D., Jiang, G., Zeng, F., Zhao, H., et al. (2020). Application of PSO-RBF Neural Network in Gesture Recognition of Continuous Surface EMG Signals. *Ijs* 38 (3), 2469–2480. doi:10.3233/JIFS-179535
- Yuan, J., Zhao, Z., Liu, Y., He, B., Wang, L., Xie, B., et al. (2021). DMPPT Control of Photovoltaic Microgrid Based on Improved Sparrow Search Algorithm. *IEEE Access* 9, 16623–16629. doi:10.1109/access.2021.3052960
- Yun, J., Jiang, D., Liu, Y., Sun, Y., Tao, B., et al. (2022b). Real-time Target Detection Method Based on Lightweight Convolutional Neural Network. *Front. Bioeng. Biotechnol.* 1, 1. doi:10.3389/fbioe.2022.861286
- Yun, J., Sun, Y., Li, C., Jiang, D., Tao, B., Li, G., et al. (2022a). Self-adjusting Force/bit Blending Control Based on Quantitative Factor-Scale Factor Fuzzy-PID Bit Control. *Alexandria Eng. J.* 61 (6), 4389–4397. doi:10.1016/j.aej.2021.09.067
- Zaman, H. R. R., and Gharehchopogh, F. S. (2021). An Improved Particle Swarm Optimization with Backtracking Search Optimization Algorithm for Solving Continuous Optimization Problems. *Eng. Comput.* 1, 1–35. doi:10.1007/s00366-021-01431-6
- Zhang, J., Meng, Q., Feng, X., and Shen, H. (2018). A 6-DOF Robot-Time Optimal Trajectory Planning Based on an Improved Genetic Algorithm. *Robot. Biomim.* 5 (1), 1–7. doi:10.1186/s40638-018-0085-7
- Zhao, G., Jiang, D., Liu, X., Tong, X., Sun, Y., Tao, B., et al. (2022). A Tandem Robotic Arm Inverse Kinematic Solution Based on an Improved Particle Swarm Algorithm. *Front. Bioeng. Biotech.* 1, 1. doi:10.3389/fbioe.2022.832829
- Zheng, C., Su, Y., and Müller, P. C. (2009). Simple Online Smooth Trajectory Generations for Industrial Systems. *Mechatronics* 19 (4), 571–576. doi:10.1016/j.mechatronics.2008.11.017

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Zhang, Xiao, Tong, Yun, Liu, Sun, Tao, Kong, Xu and Chen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.