KSME

Springer

Journal of Mechanical Science and Technology 35 (7) 2021

Original Article
DOI 10.1007/s12206-021-0638-5

Keywords:
· Trajectory planning
· Serial kinematic manipulator
· Quintic B-spline
· Adaptive cuckoo search algorithm

Correspondence to:
Yong Wang
ywang9868@163.com

Citation:
Zhang, L., Wang, Y., Zhao, X., Zhao, P., He, L. (2021). Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm. Journal of Mechanical Science and Technology 35 (7) (2021) 3171~3181.
http://doi.org/10.1007/s12206-021-0638-5

Received October 14th, 2020
Revised February 12th, 2021
Accepted March 31st, 2021

† Recommended by Editor
Ja Choon Koo

Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm

Lunhui Zhang, Yong Wang, Xiaoyong Zhao, Ping Zhao and Liangguo He

School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China

Abstract Feeding and unloading operation for part manufacturing are widely applied by industrial robots. In this paper, a set of algorithms has been used to reach higher efficiency and automation of trajectory planning for a 6-DOF integrated serial kinematic manipulator. Depending on the key nodes of the joint angles calculated by a hybrid inverse kinematics algorithm, the continuous quintic B-spline curve algorithm was utilized for planning smooth trajectories of the feeding motion from peer to peer. An adaptive cuckoo search (ACS) algorithm with high efficiency and excellent stability was proposed to minimize the total motion time under strict dynamic constraints. Comparing with 5 commonly used heuristic methods, the ACS algorithm has faster convergence speed and higher accuracy based on the same fitness function. To verify the implementation effect of the strategy, a 1:5 scale experimental platform was designed and built to implement the time-optimal trajectories. The simulations and experiments indicate that these algorithms lead to efficient planning of time-optimal and smooth trajectory in joint space.

1. Introduction

CNC machine tools are now widely equipped with industrial manipulators as auxiliary devices [1]. Various types of material-conveying manipulators installed in a CNC machine require a set of reasonable trajectory planning algorithms to complete the loading and unloading motions accurately and efficiently. With the trends of integration and unification of the main mechanical modules and their auxiliary devices (as shown in Fig. 1), matching issues must be addressed, such as simplifying the calculation process and improving the efficiency of ancillary work.

A plenty of optimization goals of industrial robots have received attention, including the motion time minimization, joint torque [2], time-jerk double objectives [3], time-energy orientation [4], energy saving orientation [5], mechanical life [6], running stability [7] and so on. The methods of trajectory planning, such as Bézier curve, B-spline curve, polynomial curve, higher-order trajectory, etc. are the fundamental works, and usually provide excellent kinematic characteristics. The purpose of this paper is to plan a smooth and time-optimal trajectory efficiently for serial manipulator. Reducing the computational burden is another kind of viable objective to improve the trajectory planning efficiency [8].

Inspired by the behavior of biological phenomena, the genetic algorithm (GA) was proposed by Holland [9] in 1967 and was subsequently improved by many works. It has been widely used in many fields, such as automation and robotics, especially in solving nonlinear and multi-objective optimization problems. A number of new evolutionary algorithms have been proposed after the GA, such as particle swarm optimization [10], ant colony optimization [11], firefly algorithm [12], cuckoo search algorithm [13] and so on. Dong et al. [14] proposed an improved immune clonal selection algorithm (IICAS) to optimized the total motion time and the energy consumption. Feng and Cheng [15] compared higher order polynomials, with trapezoid, cycloid and zero acceleration trajectories. A point-to-point polynomials trajectory with minimum absolute input energy was improved by a GA coding in the real number. Particle swarm optimization

Fig. 1. Integrated system of CNC machine tool and mechanical manipulator.

(PSO) also is an evolutionary computation invented by Eberhart and Kennedy [10] in 1995. They were inspired by the phenomenon of birds flock's looking for food. In the research field of robotics optimization, Yao et al. [16] established a minimum time optimization model constrained by the speed limitation. Roberge et al. [17] used GA and PSO to compute feasible and quasi-optimal trajectories for unmanned aerial vehicles (UAVs). PSO method has many improved branches such as adding adaptive learning factors, changing inertia weight factors and adjusting other adjustable parameters.

In recent research of heuristic algorithms, cuckoo search (CS) algorithm was first developed by Yang and Deb [13]. This algorithm is famous for the global search ability. Panda [18] maximized the work volume of a 3R manipulator by an adaptive CS algorithm. Zhang et al. [19] used a hybrid multi-objective CS algorithm to solve multi-objective optimization problems. The local search ability of the method has been enhanced. All these heuristic algorithms and their improved methods discussed above have the similar features as following: (1) they have a certain number of evolution population or searching elements; (2) they all need an evaluation function; (3) they all run through a calculation loop to improve the fitness of population.

The main contents of this paper are organized as follows. A trajectory planning scheme of a serial kinematic manipulator (SKM) integrated into a CNC machine tool is taken as the research object. The fundamental idea is to simplify the whole process of trajectory programing and achieve smooth routes and time-optimal performances of industrial robots. The geometric method was combined with an algebraic method to calculate the joint angle, for making use of their respective advantages. After obtaining the key points in joint space, a peer-to-peer cubic and quintic B-spline curve paths trajectory were planned that ignore obstacle avoidance. Under angular velocity, angular acceleration, and angular jerk constraints, an adaptive cuckoo search algorithm plays a key role in minimizing the motion time of the B-spline curves. A smooth and time-optimal trajectory was designed in the joint coordinate system. All these simulation results were transferred to the microprocessor in a scaled-down three-degree-of-freedom (3-DOF) experimental platform for verification. Furthermore, numerical simulation and experimental data are compared, demonstrating the effectiveness of these algorithms.

Table 1. Connecting rod parameters of industrial robots.

Joint	θ_i	d_i (cm)	a_i (cm)	α_i (°)
i = 1	θ_1	65	−50	−90
i = 2	θ_2	0	150	0
i = 3	θ_3	0	60	−90
i = 4	θ_4	180	0	90
i = 5	θ_5	0	0	−90
i = 6	θ_6	20	0	0

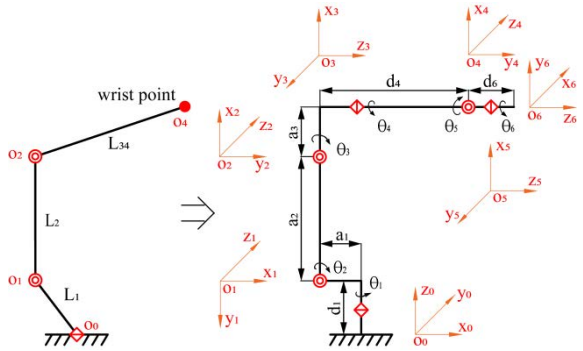

Fig. 2. The kinematic model of the 6-DOF SKM.

The first contribution of this paper is an ACS algorithm, which can converge to the best optimized solution rapidly. The second main contribution is a reliable scheme that can plan a smooth trajectory with time-minimum performance and strict dynamic constricts. Moreover, this strategy can be easily applied to other similar engineering problem.

2. Trajectory planning for SKM

2.1 Hybrid strategy for inverse kinematics analysis

A 6-DOF serial industrial manipulator integrated in CNC machine tool was design as a research object, whose Denavit–Hartenberg (D-H) parameters are shown in Table 1. For its special proportion of the structure parameters, the device can be folded up to saving space when it shut down.

In particular, the structure of serial manipulator can be transformed into 3-DOF simplified diagram, by viewing the link 3 and link 4 as one link, as is shown in left picture of Fig. 2. The rotation center of joint 5 coincides with the end point (the wrist point) of the theoretical 3-DOF model. Combining the versatility of algebraic method and accuracy of geometric method, a hybrid inverse kinematics strategy is used herein to increase efficiency of solving posture angles in SKM.

In the inverse kinematic analysis, the solution process of the 6-DOF manipulator can be divided into two steps.

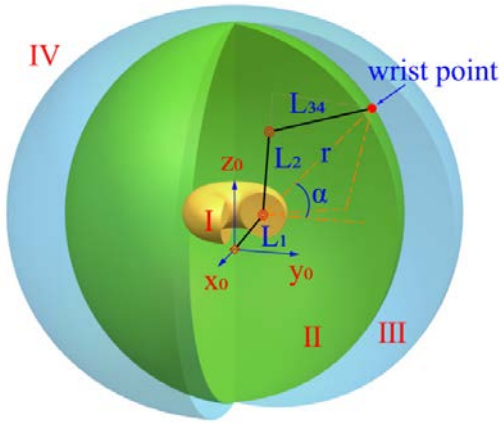

Fig. 3. The sectional view of solution sets regions.

(1) Geometric method

Since the axes of the last three joints intersect at one point (the wrist point), which satisfies the Pieper criterion, the first four arms can be evolved into a three-joint manipulator. In particular, the third and fourth arms are fitted into a link L_{34} starting from O_2 to O_4, as shown in Fig. 2.

A 4×4 pose matrix ${}^{A}_{B}\boldsymbol{T}$ composed of the rotation and translation information can be calculated depending on the end-effector on the initial coordinate system O-$x_Ay_Az_A$:

$$ {}^{A}_{B}\boldsymbol{T} = \begin{bmatrix} n_x & o_x & \alpha_x & p_x \\ n_y & o_y & \alpha_y & p_y \\ n_z & o_z & \alpha_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

where n_i, o_i, and α_i are the spatial angle relations between the terminal coordinate system O-$x_By_Bz_B$ and the initial coordinate system O-$x_Ay_Az_A$, as shown in Fig. 2, and p_i relates the coordinate of the end-effector to the initial coordinate system. A geometric method is proposed to determine the values of the first three joint angles θ_1, θ_2 and θ_3. The coordinates of the wrist point can be expressed as follows:

$$ x = p_x - d_6\alpha_x,\ y = p_y - d_6\alpha_y,\ z = p_z - d_6\alpha_z . \tag{2} $$

If the wrist point satisfies the following inequalities, it will exist in the intersection region II, as shown in Fig. 3. θ_1, θ_2 and θ_3 will have four sets of solutions. Specially, the theoretical motion range of the wrist point ignores the physical interference of devices. Some unreasonable solutions will be excluded later.

$$ \begin{cases} x^2 + y^2 < \left\{ \left[\left(L_2 + L_{34} \right)^2 - \left(z - d_1 \right)^2 \right]^{1/2} - a_1 \right\}^2 \\ d_1 - \left[\left(L_2 + L_{34} \right)^2 - a_1^2 \right]^{\frac{1}{2}} < z \\ d_1 + \left[\left(L_2 + L_{34} \right)^2 - a_1^2 \right]^{\frac{1}{2}} > z \\ x^2 + y^2 + z^2 - 2zd_1 - 2a_1\left(x^2 + y^2 \right)^{\frac{1}{2}} + a_1^2 + d_1^2 > \left(L_2 - L_{34} \right)^2 \end{cases} \tag{3} $$

where

$$ L_2 = a_2,\ L_{34} = \left(a_3^2 + d_4^2 \right)^{\frac{1}{2}} . \tag{4} $$

When the wrist point is outside the region II, then θ_1, θ_2 and θ_3 have three sets (on the II−III interface), two sets (in the region III), one set (on the I−II interface and III−IV interface), or zero sets (in the region I and IV) of solutions. In the following sections, we mainly focus on the case with four sets of solutions.

Two solutions of θ_1 can be expressed as follows:

$$ \begin{cases} \theta_{1\_1} = \arctan\left(\dfrac{y}{x} \right) \\ \theta_{1\_2} = \arctan\left(\dfrac{y}{x} \right) + \pi \end{cases} \quad (-\pi/2 < \theta_{1\_1} < \pi/2) . \tag{5} $$

For each two joints result θ_1, the two positions of link L_1 are symmetric about the z-axis in the initial coordinate system, as depicted in Fig. 3. To keep the wrist point at the same spatial coordinate, θ_2 and θ_3 must be adjusted flexibly to achieve this pose.

If θ_1 is fixed at a constant value, θ_2 and θ_3, which are interrelated variables, also have two sets of solutions. The two results of θ_2 are expressed as follows:

$$ \begin{cases} \theta_{2\_1} = -\arctan\left(\dfrac{z - d_1}{\left(x^2 + y^2 \right)^{1/2} - a_1} \right) - \alpha \\ \theta_{2\_2} = -\arctan\left(\dfrac{z - d_1}{\left(x^2 + y^2 - a_1 \right)^{1/2}} \right) + \alpha \end{cases} \tag{6} $$

where

$$ \alpha = \arccos\left(a_2^2 + r^2 - \frac{a_3^2 + a_4^2}{2a_2 r} \right) . $$

In the same way, each θ_2 solution determines a θ_3 solution, as follows:

$$ \begin{cases} \theta_{3\_1} = -\arctan\left(\dfrac{d_4}{a_3} \right) + \pi - \arccos\left(\dfrac{a_2^2 + a_3^2 + a_4^2 - r^2}{2a_2\left(a_3^2 + a_4^2 \right)^{1/2}} \right) \\ \theta_{3\_2} = -\arctan\left(\dfrac{d_4}{a_3} \right) - \pi + \arccos\left(\dfrac{a_2^2 + a_3^2 + a_4^2 - r^2}{2a_2\left(a_3^2 + a_4^2 \right)^{1/2}} \right) \end{cases} \tag{7} $$

where

$$ r = \left\{ \left[\left(x^2 + y^2 \right)^{1/2} + a_1 \right]^2 + \left(z - d_1 \right)^2 \right\}^{1/2} . $$

Within the range given by Eq. (3), each wrist point coordinate has two solutions for θ_1, and the solution of each θ_1 also corresponds to the solutions of two sets of θ_2 and θ_3. In the four scenarios, there are unreasonable solutions that make the posture of the manipulator impossible to achieve.

(2) Algebraic method

We invert the first three sets of the relative pose matrices ${}^{A}_{B}\boldsymbol{T}$ (*A*, *B* = 0, 1, 2, 3, 4, 5, 6), and the relationship between θ_1, θ_2, θ_3 and θ_4, θ_5, θ_6 is as follows:

$$ {}^{3}_{2}\boldsymbol{T}(\theta_3)\,{}^{2}_{1}\boldsymbol{T}(\theta_2)\,{}^{1}_{0}\boldsymbol{T}(\theta_1)\,{}^{0}_{6}\boldsymbol{T} = {}^{3}_{4}\boldsymbol{T}(\theta_4)\,{}^{4}_{5}\boldsymbol{T}(\theta_5)\,{}^{5}_{6}\boldsymbol{T}(\theta_6) \tag{8} $$

where

$$ {}^{A}_{B}\boldsymbol{T}^{-1} = {}^{B}_{A}\boldsymbol{T} . $$

Inserting the values of θ_1, θ_2, θ_3 obtained by the inverse method presented above into Eq. (8), θ_4, θ_5, θ_6 can determined. On the right side of Eq. (8), the parameter matrix can be expressed as follows:

There are two solutions of θ_4, expressed as follows:

$$ \begin{cases} \theta_{4\_1} = \arctan\left(\dfrac{{}^{3}_{2}\boldsymbol{T}(\theta_3)\,{}^{2}_{1}\boldsymbol{T}(\theta_2)\,{}^{1}_{0}\boldsymbol{T}(\theta_1)\,{}^{0}_{6}\boldsymbol{T}(2,3)}{{}^{3}_{2}\boldsymbol{T}(\theta_3)\,{}^{2}_{1}\boldsymbol{T}(\theta_2)\,{}^{1}_{0}\boldsymbol{T}(\theta_1)\,{}^{0}_{6}\boldsymbol{T}(1,3)}\right) \\ \theta_{4\_2} = \theta_{4\_1} + \pi \end{cases} . \tag{9} $$

Furthermore, each θ_4 corresponds to a θ_5, and each group of θ_4 and θ_5 corresponds to a unique θ_6:

$$ \theta_5 = \arcsin\left(-\frac{{}^{3}_{2}\boldsymbol{T}(\theta_3)\,{}^{2}_{1}\boldsymbol{T}(\theta_2)\,{}^{1}_{0}\boldsymbol{T}(\theta_1)\,{}^{0}_{6}\boldsymbol{T}(2,3)}{\sin\theta_4}\right), \tag{10} $$

$$ \theta_6 = \arccos\left(\frac{{}^{3}_{2}\boldsymbol{T}(\theta_3)\,{}^{2}_{1}\boldsymbol{T}(\theta_2)\,{}^{1}_{0}\boldsymbol{T}(\theta_1)\,{}^{0}_{6}\boldsymbol{T}(3,1)}{\sin\theta_5}\right). \tag{11} $$

After obtaining θ_4, there are many possible values of θ_5 and θ_6, but the correct and reasonable solution always ensures that the elements of the matrix equation on both sides correspond to each other, which can be used to screen the results.

2.2 Construction of cubic and quintic B-spline interpolation function

A uniform B-spline curve avoids the shortcomings of Bézier curves, whose control points affect the whole curve, while mostly retaining the Bézier curve's advantage of being continuous and smooth. When the control points are changed, only the local trajectory of the B-spline curve is affected, yet the remaining segment paths are fixed [20]. In addition, common interpolation curves cannot accurately represent complex curves. If the accuracy is to be guaranteed, the order of the polynomial curve will reach a high level and introduce a large calculation load [21]. Comparing with lower-order B-spline curve, such as cubic B-spline curve, quintic B-spline curve can provide more controllable dynamics characteristics, although it requires larger computational resources and highly time-consuming calculations. In particular, quintic B-spline curve whose jerk have continuity can ensure that its starting and ending acceleration are zero, simultaneously retaining the fairing of cubic B-spline curve.

A series of data points $P_0, P_1, \ldots, P_n$ were set up on the expected trajectory in the joint space. These data points divide the B-spline trajectory into *n* segments. For a B-spline curve of degree *k*, the *i*'th segment spline can be controlled by (*k*+1) control points. According to the nature of the B-spline curve, changing the control point only affects the local trajectory around it, and the rest of the trajectory remains unchanged. The basic expression of the *i*'th section B-spline is presented as follows [22]:

$$ \boldsymbol{p}(u) = \sum_{i=0}^{n} \boldsymbol{d}_i N_{i,k}(u) \tag{12} $$

where $\boldsymbol{d}_i$ is the displacement vector of the control point, and $N_{i,k}$ (u) represents the calibration B-spline basis functions, whose first subscript *i* indicates the sequence and whose second subscript *k* represents the degree of the curve. With the parameter *u* ranging from 0 to 1, all points of the segment can be calculated by substituting *u* into Eq. (12). However, as the unobservable control points $\boldsymbol{d}_i$ in Eq. (12) are unknown, the control points in the open cubic and quintic B-spline are rendered using the following equations:

$$ \begin{bmatrix} 1 & 0 & -1 & & & & & & \\ 1 & 4 & 1 & & & & & & \\ & 1 & 4 & 1 & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \\ & & & & & 1 & 4 & 1 & \\ & & & & & 1 & 0 & -1 & \end{bmatrix} \begin{bmatrix} V_{i-1} \\ V_i \\ V_{i+1} \\ \cdots \\ V_{i+n} \\ V_{i+n+1} \end{bmatrix} = 6 \begin{bmatrix} 0 \\ P_0 \\ P_1 \\ \cdots \\ P_n \\ 0 \end{bmatrix}, \tag{13} $$

$$ \begin{bmatrix} 1 & 2 & -6 & 2 & 1 & & & & \\ 1 & 10 & 0 & -10 & -1 & & & & \\ 1 & 26 & 66 & 26 & 1 & & & & \\ & 1 & 26 & 66 & 26 & 1 & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & 1 & 26 & 66 & 26 & 1 \\ & & & & -1 & -10 & 0 & 10 & 1 \\ & & & & 1 & 2 & -6 & 2 & 1 \end{bmatrix} \begin{bmatrix} V_{i-2} \\ V_{i-1} \\ V_i \\ V_{i+1} \\ \cdots \\ V_{i+n} \\ V_{i+n+1} \\ V_{i+n+2} \end{bmatrix} = 120 \begin{bmatrix} 0 \\ 0 \\ P_0 \\ P_1 \\ \cdots \\ P_n \\ 0 \\ 0 \end{bmatrix} . \tag{14} $$

By analyzing the practical dynamic law, the first and last few laws of linear equality constraints in the previous two matrix equations are given by angular velocity and angular acceleration boundary conditions. Since the initial and final segments need the extra uniform control points, some virtual data points are put into column of data points. During actual calculations,

the basis function $N_{i,k}(u)$ in Eq. (12) can be transformed into a matrix expression of the B-spline curve, expressed as Eq. (15). The control points generated by Eqs. (13) and (14) are substituted in the following Eq. (15) to get the path points.

$$Q_i(u)=\frac{1}{k!}\begin{bmatrix}1 & u & \cdots & u^k\end{bmatrix}\boldsymbol{M}_k\begin{bmatrix}V_{i-k/2+1/2}\\V_{i-k/2+3/2}\\\vdots\\V_{i+k/2+1/2}\end{bmatrix},\quad (k=1,3,5,7,\ldots). \tag{15}$$

For instance, the *i*'th segment of the quintic B-spline curve is controlled by six control points, V_{i-2}, V_{i-1}, V_i, V_{i+1}, V_{i+2}, and V_{i+3}. The coefficient matrices $\boldsymbol{M}_k$ are given by

$$\boldsymbol{M}_3=\begin{bmatrix}1&4&1&0\\-3&0&3&0\\3&-6&3&0\\-1&3&-3&1\end{bmatrix},$$

$$\boldsymbol{M}_5=\begin{bmatrix}1&26&66&26&1&0\\-5&-50&0&50&5&0\\10&20&-60&20&10&0\\-10&20&0&-20&10&0\\5&-20&30&-20&5&0\\-1&5&-10&10&-5&1\end{bmatrix}.$$

In order to guarantee the flexibility of motion process, the trajectory performances of the cubic and quintic B-spline curves are compared with each other through a digital case, as presented in the next section.

2.3 Simulation of trajectory planning

In this section, a complete feeding process of serial manipulator whose D-H parameters are mentioned above are presented as a case. The end-effector of the posture provide a pose matrix ${}^0_6\boldsymbol{T}$ when manipulator finish the motion, as shown in Eq. (16). Accordingly, the angle of each joints can further be calculated. As expected, within the range specified by the Eq. (3), θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6 have eight different sets of solutions, as summarized in Table 2.

$${}^0_6\boldsymbol{T}=\begin{bmatrix}0&0&-1&-268.202\\0&1&0&0\\1&0&0&217.349\\0&0&0&1\end{bmatrix}. \tag{16}$$

The rationality of the eight sets of inverse solutions should be verified during the calculation. After rule out some unreasonable results, the only correct group of solutions can be automatically filtered out. The algorithm analyzed in this section provides a data foundation for subsequent path trajectory planning.

Table 2. Joint angle of pose matrix ***T***.

No.	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 (°)
1	**180.0**	**−38.0**	**−52.0**	**0.0**	**0.0**	**0.0**
2	180.0	−38.0	−52.0	180.0	0.0	180.0
3	180.0	−16.1	−91.1	0.0	17.3	0.0
4	180.0	−16.1	−91.1	180.0	−17.3	0.0
5	0.0	−93.2	−157.5	0.0	−19.2	180.0
6	0.0	−93.2	−157.5	180.0	19.2	0.0
7	0.0	168.3	14.4	0.0	267.3	180.0
8	0.0	168.3	14.4	180.0	92.8	0.0

Table 3. Data points of the first three joints.

Via point	Joint 1	Joint 2	Joint 3
P0 (°)	0	−60	40
P1 (°)	20	−75	50
P2 (°)	40	−110	70
P3 (°)	90	−85	40
P4 (°)	130	-60	10
P5 (°)	160	-45	-15
P6 (°)	180	-38	-52

Inserting key points between the initial angle and the final angle can adjust the range of motion and movement path. In this example, five intermediate points divide each joints trajectory into six sections, as shown in Table 3. Specifically, the second and third joints will implement reverse rotation in the early feeding motion. It could keep a posture with low moment of inertia when the first joint rotates at a high speed that the manipulator can work steadily.

The key data points are translated into (7+*k*-1) control points by Eqs. (13) and (14), then assigning the control points to Eq. (15) to fill the smooth B-spline trajectory. The cubic B-spline curve for joint 1 and quintic B-spline curve for the first three joints describing the angle, angular velocity, angular acceleration, and angular jerk of the motion are depicted in Figs. 4 and 5, respectively. The total time of these non-optimized simulations was 12 s.

The comparison between Fig. 4 and the first diagram in Fig. 5 shows that the angle and angular velocity results for the cubic and quintic curve were very similar to each other. The primary differences between the two diagrams were the initial and final states of the angular acceleration and angular jerk. Fig. 4 illustrates cubic curve only followed an abrupt stop/start and discontinuous characteristic of angular jerk. The first picture in Fig. 5 show that the maximum absolute value of the quintic angular velocity only reached 29.12°/s, and the maximum absolute value of the angular acceleration only reached 15.76°/s^2.

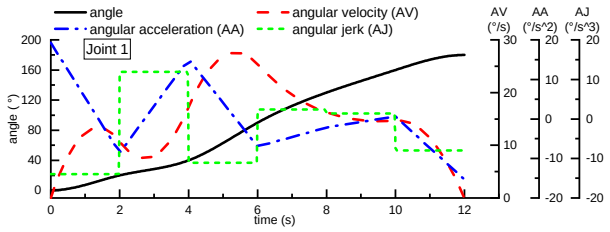

Fig. 4. Unoptimized cubic B-spline curve for joint 1.

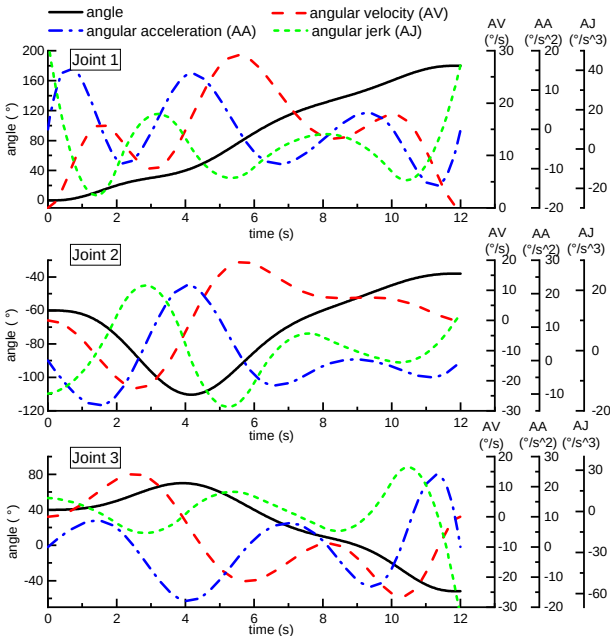

Fig. 5. Unoptimized quintic B-spline curve for three joints.

Therefore, the quintic B-spline curve is a better method compared to cubic one, for which greatly reduce the vibration of the equipment and improve the stability of auxiliary operation. In particular, the angular jerk in quintic graph has a wider range of variation. Thus, the total time of uniform B-spline curve could be adjusted for optimization.

3. Time optimization algorithm

In this section, we present a unified fitness function as the only criterion to evaluate the of objective individual performance. Then, the ACS algorithm and other five different heuristic algorithms are to optimize each pair of adjacent knots and minimize the entire time. The rationale of our strategy is as follows.

3.1 Mathematical model

1) Dynamic constraints

By calculating the first, second and third derivatives of quintic B-spline function $Q_i(u)$, we obtain the angular velocity, angular acceleration and angular jerk expressions presented below:

$$\dot{\theta}_i(u)=\frac{d\theta_i(u)}{du}\frac{du}{dt}=\frac{q'}{t'}, \tag{17}$$

$$\ddot{\theta}_i(u)=\frac{d\dot{\theta}_i(u)}{dt}=\frac{d\dot{\theta}_i(u)}{du}\frac{du}{dt}=\frac{q''t'-q't''}{t'^3}, \tag{18}$$

$$\dddot{\theta}_i(u)=\frac{d\ddot{\theta}_i(u)}{du}\frac{du}{dt}=\frac{q'''t'^2+3q't''^2-3q''t't''-q't't'''}{t'^5} \tag{19}$$

where the data points of angle and time are defined as q and t, respectively. The physical devices always have limiting dynamics performance according to their structural characteristics and drivers. With the change of time intervals, the first, second and third derivatives of quintic B-spline curves sometimes exceed the limit value, but they always maintain the continuity. Besides the two boundary points, each segment in these curves may have a unique extreme point or minima point. The value of boundary $|\dot{\theta}_i|$, $|\ddot{\theta}_i|$, $|\dddot{\theta}_i|$ in i'th segment are calculated and compared with the extreme value $|\dot{\theta}_{ix}|$, $|\ddot{\theta}_{ix}|$, $|\dddot{\theta}_{ix}|$, the larger one would be given strict constraints which satisfy the following Eq. (20). Additionally, extreme value $|\dot{\theta}_{ix}|$, $|\ddot{\theta}_{ix}|$, $|\dddot{\theta}_{ix}|$ are solved by the golden-section method.

$$\begin{cases}\max\left\{|\dot{\theta}_i|,|\dot{\theta}_{ix}|\right\}\le\dot{\theta}_{\max}\\ \max\left\{|\ddot{\theta}_i|,|\ddot{\theta}_{ix}|\right\}\le\ddot{\theta}_{\max}\\ \max\left\{|\dddot{\theta}_i|,|\dddot{\theta}_{ix}|\right\}\le\dddot{\theta}_{\max}\end{cases} \tag{20}$$

where $\dot{\theta}_{\max}$, $\ddot{\theta}_{\max}$ and $\dddot{\theta}_{\max}$ are the maximum values of angular velocity, angular acceleration and angular jerk.

(2) Objective function

The objective function is built as follow according to the constraints defined by Eq. (20)

$$F_{obj}=\sum X_i+a_1\sum v_e+a_2\sum a_e+a_3\sum j_e$$
$$(i=1,2,\cdots m-1) \tag{21}$$

where X_i present the time derivative in i'th segment; v_e, a_e and j_e are the excess values of the limit angular velocity, limit angular acceleration and limit angular jerk. The last three elements are penalty functions that provide a restrict area for optimization. It is desirable in the case that penalty parts are substantially zero.

3.2 Heuristic algorithm for time optimization

The CS algorithm uses Lévy flight to simulate the behavior that cuckoo is looking for nests. The Lévy can generate the large step and small step randomly. The standard CS algorithm is expressed as:

$$X_i^{t+1}=X_i^t+\alpha\oplus\text{Lévy}(\beta) \tag{22}$$

where the Lévy step [23] is expressed as:

$$\text{Lévy}(\beta) = \frac{\mu}{|v|^{1/\beta}} \cdot \left(\frac{\Gamma(1+\beta)\cdot\sin(\pi\beta/2)}{\Gamma[(1+\beta)\cdot\beta\cdot 2^{(\beta-1)/2}/2]}\right)^{1/\beta} \quad (23)$$

where $\alpha > 0$; $\beta = 1.5$; μ and v are normal distribution, respectively.

When the nests near global best solution are found, the standard CS algorithm still keep equal step size. The large step leads to rapid search of global optimum with low accuracy. The small step has opposite effect. Hence, an adaptive coefficient α_A is built to replace α, expressed as:

$$X_i^{t+1} = X_i^t + \alpha_A \oplus \text{Lévy}(\beta) \quad (24)$$

where

$$\alpha_A = a \cdot N_p^{\ b} \exp\left(-\frac{t \cdot F_m^t}{t_{\max}}\right) \quad (25)$$

where a and b are variables; N_p is the size of initial population; $t_{\max}$ is the maximum number of iterations. During large step searching, the mean fitness value of nests will have a dramatic decline. Therefore, the mean fitness value of the t'th generation population defined as F_m^t is imported into coefficient formula. By adding these factors, the whole population can approach the global optimum quickly in the early calculation, and find the high precision optimal solution with short step in the later calculation.

The main steps of ACS are presented below:

Step 1: Set the initial population size N_p, the maximum number of iterations and other coefficients. And the fitness values of initial nests are calculated.

Step 2: Update the nests by improved Lévy flights. And update the individual fitness values of new nests.

Step 3: A percentage (*Pa*) of poor nests is rejected and the new nests are developed at new habitats via Lévy flights. Similarly, update the individual fitness values of new nests.

Step 4: The satisfaction of convergence condition is judged. If the condition is satisfied, exist loop. If not, go back to the step 2.

Step 5: Select out the best individual and output their fitness values.

In order to get its performance analysis results, five popular optimized heuristic search algorithms, the standard genetic algorithm (GA), the improved adaptive genetic algorithm (IAGA), particle swarm optimization (PSO), hybrid particle swarm optimization (HPSO) and cuckoo search (CS) were programmed to execute the same simulation. Based on the principle of optimization, the ACS algorithm will run objective function less than $2\times N_P$ times per generation. GA and IAGA only run objective function N_P times per generation. Similarly, PSO and HPSO run $2\times N_P$ times.

The only convergence condition in this research is the maximum iterative times. The initial population of these algorithms was set as 100. The range of population is 0.1 to 3. The maximum number of iterations is set as 500. a value is 0.7; b value is 0.5; *Pa* value is 0.25. All six groups of time-optimized results are presented below.

Table 4. Optimized results with different algorithms for joint 1.

Algorithm	Initial population	Best fitness	Time consumption
GA	10	10.491	31.2 s
	50	8.498	164.8 s
	100	6.766	318.5 s
	500	6.161	1652.8 s
IAGA	10	10.550	34.9 s
	50	7.411	171.0 s
	100	6.416	339.9 s
	500	6.304	2327.8 s
PSO	10	6.722	66.7 s
	50	6.338	318.2 s
	100	6.250	633.4 s
	500	6.146	3158.3 s
HPSO	10	6.469	65.1 s
	50	6.162	318.2 s
	100	6.126	634.0 s
	500	6.129	3169.6 s
CS	10	6.047	61.5 s
	50	6.069	300.2 s
	100	6.064	602.8 s
	500	6.058	3015.7 s
ACS	10	6.034	63.3 s
	50	6.045	310.7 s
	100	6.047	615.5 s
	500	6.054	3095.7 s

Compared with the standard group of algorithms, the improved group obviously increased the convergence rate. The GA and IAGA depended heavily on the size of population, are prone to low convergence speeds or falling into local optima, as is shown in Fig. 6. The lines of GA and IAGA in first figure are completely ineffective. The precision of PSO and HPSO are low in this situation. The standard CS algorithm has excellent global search ability in this case with only 10 initial population saving much computing resources. But, the final accuracy of CS algorithm needs to be improved. Not only does the big population size increase the computational burden, but also it reduces the optimization accuracy.

The results show that the ACS algorithm has a better solution accuracy and stability, while it only spends little computing time.

3.3 Optimization results

Based on the structure and maximum torque of the first three joints, 50º/s^2, 30º/s^2, and 40º/s^2, the maximum angular accel-

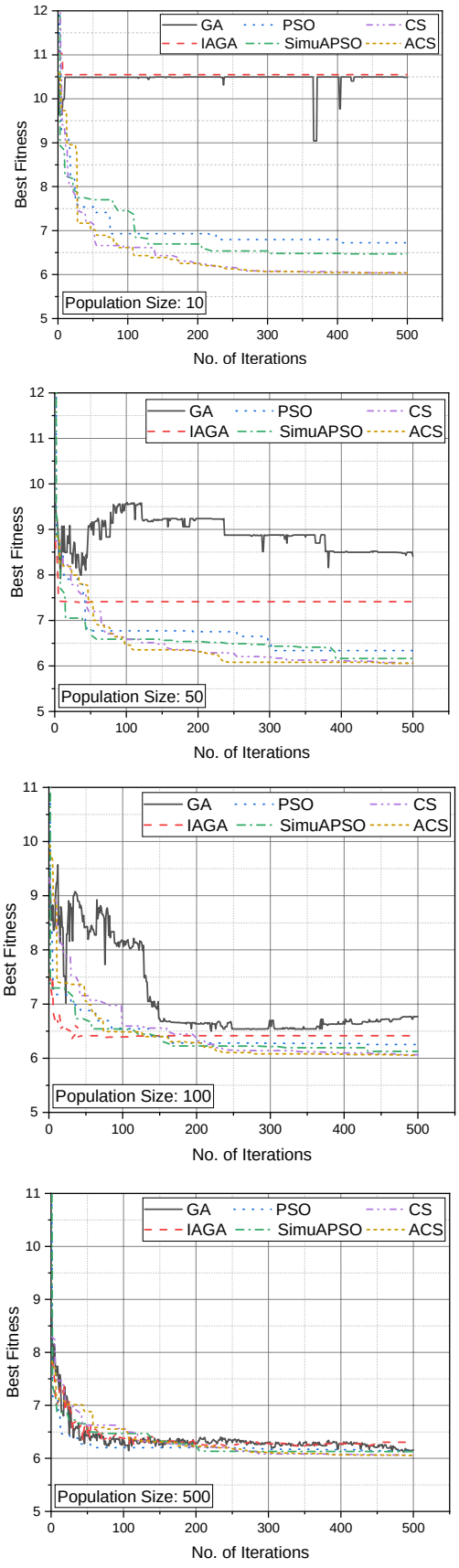

Fig. 6. Performance in different heuristic search algorithms.

Table 5. Constraint conditions of joint performance.

Joint	$\dot{\theta}_{max}$ (°/s)	$\ddot{\theta}_{max}$ (°/s^2)	$\dddot{\theta}_{max}$ (°/s^3)
Joint 1	50	50	150
Joint 2	40	30	120
Joint 3	60	40	150

Table 6. Time-optimal results.

Time period	Initial value	Joint 1	Joint 2	Joint 3
X_1 (s)	2	1.514	1.132	0.776
X_2 (s)	2	0.498	1.613	1.174
X_3 (s)	2	1.638	2.045	1.608
X_4 (s)	2	0.713	0.221	0.732
X_5 (s)	2	0.257	0.461	0.782
X_6 (s)	2	1.413	0.988	1.746
Sum (s)	12	6.033	6.460	6.818

erations are determined first. In order to have approximately the same execution time, the maximum angular velocities were set to different values. The extremal performance parameters of the first three joints are listed in Table 5. Inserting the limit values of $\dot{\theta}_{max}$, $\ddot{\theta}_{max}$ and $\dddot{\theta}_{max}$ into Eq. (20), the constraint equations were obtained and substituted into the ACS algorithms above to calculate the optimization results. The parameters for the time periods of the initial trajectory and the three optimized trajectories are listed in Table 6.

The optimized time intervals obtained above are submitted into Eq. (14) to obtain the control points in time axis. Then, the time-optimal quintic B-spline curves for the first three joints are composed of the control points in time and angle coordinates. As a result, their angle, angular velocity, angular acceleration and angular jerk curves are depicted in Fig. 7.

The minimum-time of optimized quintic curves in Fig. 7 are significantly less as compared to the unoptimized ones in Fig. 5. By comparison, we found that the first, second and third derivative curves of time-optimal quintic B-spline trajectory reach their limits in multiple points. In figure of joint 1, angular jerk become the main limiting factor of time minimization. Under the parameter constricts, joint 2 made maximum use of the torque, so that the angular velocity can reach two opposite extreme values in the shortest time. Specifically, the total travel-time of the three joints was reduced by about 46.4 % compared to that of the non-optimized manipulator. Due to time intervals optimization and B-spline, the end of serial kinematic manipulator went through a smoother and time-optimal path.

4. Experiments

4.1 Design and implementation of experiment in 3-DOF manipulator platform

Comparisons with experiments play a vital role in bridging

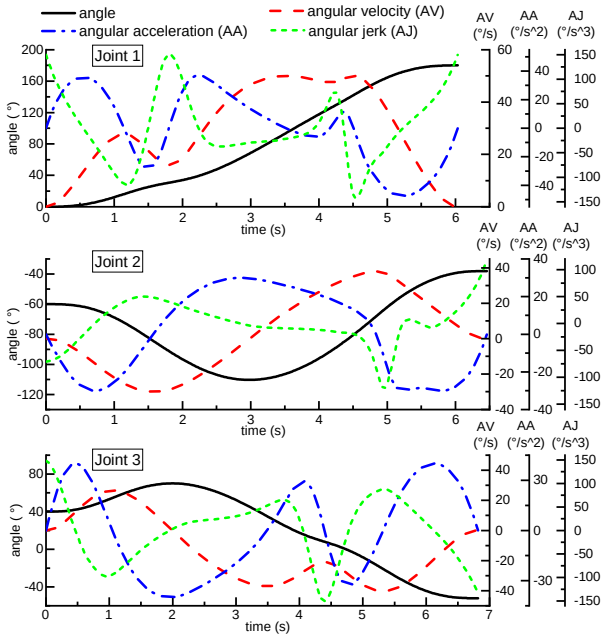

Fig. 7. Optimized motion state curve of three joints.

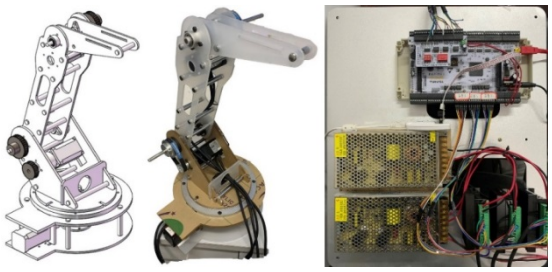

Fig. 8. 3-DOF design model of SKM (left); experimental SKM (mid); control platform (right).

the gap between theory and practice. To analyze the feasibility of the three algorithms above, a 3-DOF SKM platform was designed and built for verification. A 3D design model, physical 3-DOF manipulator, and integrated control platform are presented in Fig. 8.

The hardware section has two structural features. The first feature of the manipulator is that the experimental 3-DOF manipulator was established in a 1/5 scaled-down size for convenience. To ensure each joint can rotate under a normal load, the dynamic and static characteristics of 3-DOF SKM were examined. Due to the torque demand in the second joint, it was necessary to add a reduction gearhead with a reduction ratio of 1/10 to increase the output torque. The base rotary joint was transmitted by a 1/2 bevel gear to change the torque vectoring. The second feature is that each linkage was driven by a stepper motor, which varied the electrical pulses for the angular displacement. Therefore, the microprocessor, DC power supply, and motor drives were installed on the control platform to send pulses to the motors. To collect the experimental data, each stepper motor was equipped with a rotary incremental encoder to measure the rising and falling edges of the pulse

Table 7. Configuration information of each joint.

Device parameter	Joint 1 (bottom)	Joint 2 (middle)	Joint 3 (top)
Motor torque	0.65 N · m	0.65 N · m	0.4 N · m
Subdivision of driver	32	8	32
Angle per step	0.05625°	0.225°	0.05625°
Gear ratio	1:2	1:20	1:2

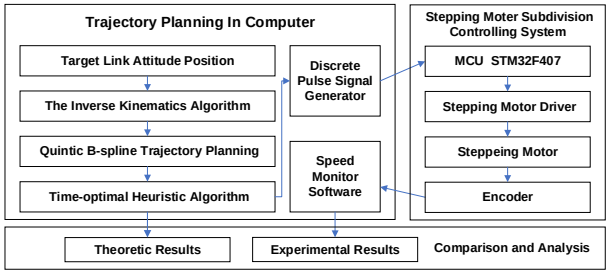

Fig. 9. Diagram of the manipulator trajectory planning system.

from the rotating disk. The configuration information of the 3-DOF SKM is listed in Table 7.

The software section mainly includes a quantitative rotation controller and an angular pulse measurer. The theoretical angular displacement data was converted to a pulse signal that the motor could recognize, while the time-varying pulse could control the motor rotation speed for completion of the expected speed control. Such a demand requires the output compare mode of the advanced timer in the STM32F407 to output a time-varying pulse signal. After transformation calculations, the discrete impulse frequency of predetermined motion was burned into the microprocessor, controlling the three stepper motors to actuate the 3-DOF manipulator. Data acquisition is another important task. Specifically, the rotary encoder mounted on the joint generated 4000 pulsing signals for every one rotation of shaft axis. During the operation of the manipulator, the encoder could measure the angular displacement immediately to transmit the angular and time information to the PC for analysis. The whole operation of the theoretical calculation and verification of the experiments is shown in the operational flow diagram in Fig. 9.

4.2 Collection and analysis for experimental data

Fig. 10 shows the initial and final postures of the feeding motion in both the simulation and experimental manipulator models. The motions of the 3-DOF experimental platform were implemented primarily in order to verify the motion simulation results in numerical model. The feeding motion can be simply described as picking the workpiece and delivering it to the installation position.

The output pulse counts of three motors were generated by the encoders every 0.1 s. To determine the reliability of the

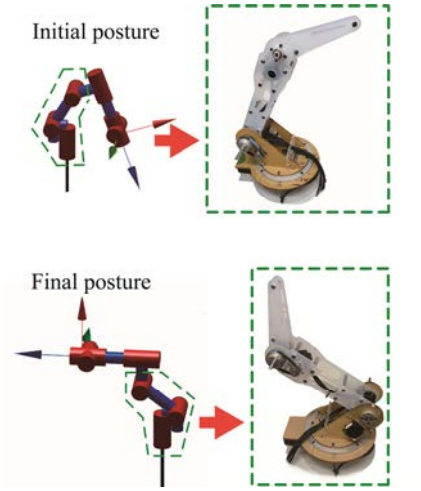

Fig. 10. Postures transition of numerical model and experimental model.

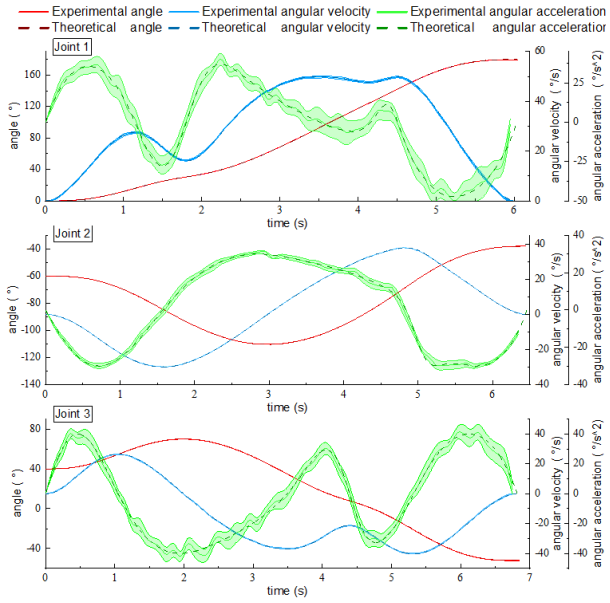

Fig. 11. Theoretical and experimental results on time-optimal quintic B-spline curves.

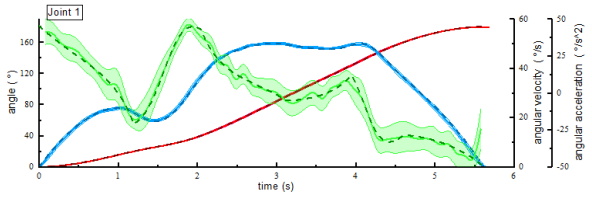

Fig. 12. Theoretical and experimental results on time-optimal cubic B-spline curves.

dynamic information, both the time-optimal quintic and cubic B-spline motions were implemented ten times.

All groups of information from upper-monitor were sent to a transformation program for plotting the angle-time, angluar velocity–time, and angular acceleration-time curves. Moreover, the mean curve could be obtained through further processing. Fig. 11 shows the time-optimal quintic B-spline curves results, and Fig. 12 shows the experimental results of the first joint following the time-optimal cubic B-spline curve. In these drawing, the solid lines in the plot are the mean curves of the kinematic data, and the translucent areas around the solid lines represent the standard error of the mean (SEM), which indicates the experimental error and uncertainty. The theoretical curves are plotted as dotted lines.

A comparison of the results revealed the following:

1) Based on the angle curves in all four diagrams, the experimental trajectory curves in this example almost coincided with the theoretical curves, and their error bars were acceptable. Furthermore, SEM of the three angle curves is very small.

2) Some deviations appeared in the mean curves of the angular velocity and acceleration compared with the theoretical curves. The initial and final regions of the angular velocity curves in Fig. 12 lag behind the theoretical curve evidently. But quintic B-spline curve doesn't have that problem.

3) After time optimization, the angular velocity and acceleration of quintic curves still satisfied their constraint requirements. Though quintic curves had the better starting and stopping performances, the running time of cubic curve was 6.9 % less than that of quintic one in this case. Moreover, the angular velocity curve of cubic one had an acceptable result. It indicates that the cubic trajectory is available at low-load and low-precision situation.

To sum up, the quintic B-spline curve has a better acceleration performance than the cubic B-spline curve, for which the angular acceleration error of the was reduced and the system robustness increased. The experimental results demonstrated the reliability of the time-optimal trajectory planning scheme.

5. Conclusions

In this paper, a series of trajectory planning and optimization algorithms are proposed for an auxiliary manipulator integrated in CNC equipment. Given consideration to the same dynamic constraints, an adaptive cuckoo search algorithm and other heuristics were utilized to minimize the same trajectory. By comparison, the results indicate that the ACS algorithm has an excellent global searching ability and fast local convergence rate, which is suitable for multivariable optimal solution search problems. The total time of quintic B-spline curve for feeding motion was approximately reduced by 28 %. In addition, the practicability of a smooth and time-optimal trajectory was verified using a 3-DOF experimental platform. The scheme can plan smooth and time-optimal trajectory in joint space efficiently and can be widely applied to other manipulators.

In future work, we will improve the scheme and algorithms presented in this paper. Supplementary issues such as obstacle avoidance and anti-collision, are further taken into account, will be take into account for industrial applications of machine–robot integrated systems.

Acknowledgements

This research was supported financially by the Anhui Province Major Research and Development Project (Grant No. 16030901018). The design and testing work presented in this paper are some of the results from a cooperative research effort. The authors would like to acknowledge the valuable discussions and suggestions from the members of the group.

References

[1] J. Wan, H. T. Wu, R. Ma and L. A. Zhang, A study on avoiding joint limits for inverse kinematics of redundant manipulators using improved clamping weighted least-norm method, *Journal of Mechanical Science and Technology*, 32 (2018) 1367-1378.

[2] X. Gao, M. T. Du, L. Zhai, Y. H. Zhang, H. X. Sun, X. P. Diao, L. Long, Y. Li, X. Li and Y. Dai, Research on joint torque optimization method of redundant space manipulators with vibration suppression, *ITM Web of Conferences*, 12 (2017) 01004.

[3] A. Gasparetto, A. Lanzutti, R. Vidoni and V. Zanotto, Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning, *Robotics and Computer-Integrated Manufacturing*, 28 (2012) 164-181.

[4] C.-T. Chen and T.-T. Liao, A hybrid strategy for the time- and energy-efficient trajectory planning of parallel platform manipulators, *Robotics and Computer-Integrated Manufacturing*, 27 (2011) 72-81.

[5] R.-F. Fung and Y.-H. Cheng, Trajectory planning based on minimum absolute input energy for an LCD glass-handling robot, *Applied Mathematical Modelling*, 38 (2014) 2837-2847.

[6] L. Cui, H. Wang and W. Chen, Trajectory planning of a spatial flexible manipulator for vibration suppression, *Robotics and Autonomous Systems*, 123 (2020).

[7] J. Kim and W. K. Chung, Real-time zero moment point compensation method using null motion for mobile manipulators, *Advanced Robotics*, 20 (2006) 581-593.

[8] H. Wang, H. Wang, J. Huang, B. Zhao and L. Quan, Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve, *Mech. Mach. Theory*, 139 (2019) 284-293.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge (1992).

[10] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of ICNN'95 - International Conference on Neural Networks*, 4 (1995) 1942-1948.

[11] F. Li, J. Zhao, X. Song, P. Zhou, S. Fang and Z. Liu, Path planning of 6-DOF humanoid manipulator based on improved ant colony algorithm, *Proc. of the 24th Chinese Control and Decision Conference (CCDC)* (2012) 4158-4161.

[12] M. Mavrovouniotis, C. Li and S. Yang, A survey of swarm intelligence for dynamic optimization: algorithms and applications, *Swarm and Evolutionary Computation*, 33 (2017) 1-17.

[13] X. S. Yang and S. Deb, Cuckoo search via levy flights, *Mathematics* (2010) 210-214.

[14] C. Dong, L. Shiqi, W. J. Feng, F. Yi and L. Yang, A multi-objective trajectory planning method based on the improved immune clonal selection algorithm, *Robotics and Computer-Integrated Manufacturing*, 59 (2019) 431-442.

[15] R. F. Fung and Y. H. Cheng, Trajectory planning based on minimum absolute input energy for an LCD glass-handling robot, *Applied Mathematical Modelling,* 38 (11-12) (2014) 2837-2847.

[16] J. Yao, Y. Huang and Z. Wan, Minimum-time trajectory planning for an inchworm-like climbing robot based on quantum-behaved particle swarm optimization, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231 (18) (2017) 3443-3454.

[17] V. Roberge, M. Tarbouchi and G. Labonte, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, *IEEE Transactions on Industrial Informatics*, 9 (1) (2013) 132-141.

[18] S. Panda, D. Mishra and B. B. Biswal, An approach for design optimization of 3R manipulator using adaptive cuckoo search algorithm, *Mechanics Based Design of Structures and Machines,* 48 (6) (2020) 773-798.

[19] M. Zhang, Hybrid multi-objective cuckoo search with dynamical local search, *Memetic Computing*, 10 (2017) 1-10.

[20] L. Piegl and W. Tiller, *The NURBS Book*, Springer Science and Business Media (2012).

[21] L. Zhang, M. Nie and H. Tong, Cubic and quinitic B-spline functions for dynamic response, *J. Tsinhua Univ. (Sci. and Tech.)*, 46 (2006) 25-28.

[22] S. A. Bazaz and B. Tondu, 3-cubic spline for online Cartesian space trajectory planning of an industrial manipulator, *International Workshop on Advanced Motion Control* (1998) 493-498.

[23] X. S. Yang and S. Deb, Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1 (2010) 330-343.

Lunhui Zhang received his Bachelor's degree from Changchun University of Science and Technology in 2018. He is currently pursuing a Master's degree in Mechanical Engineering Department, Hefei University of Technology, Hefei, Anhui, P. R. China. His major research focus on design and control of robotic systems.

Yong Wang is currently a Full Professor at the Mechanical Engineering Department, Hefei University of Technology, Hefei, Anhui, P. R. China. His current research interests include intelligent robotic systems and ubiquitous ports.