Automation and Machine Learning (2024) **DOI: 10.23977/autml.2024.050109**
Clausius Scientific Press, Canada **ISSN 2516-5003 Vol. 5 Num. 1**

Robotic arm trajectory planning based on COA optimization algorithm

Guangyu Du[1,a], Su Xu[1,b,*], Chuande Xu[1,c], Wenxuan Cui[2,d]

[1]School of Electronic Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu, 222000, China
[2]School of Mechanical Engineering, Jiangsu Ocean University, Lianyungang, Jiangsu, 222000, China
[a]2023220621@jou.edu.cn, [b]1995000021@jou.edu.cn, [c]2023220630@jou.edu.cn, [d]2022210501@jou.edu.cn
[*]Corresponding author

Keywords: COA optimization algorithm, Trajectory planning, Fifth degree polynomial interpolation, MATLAB

Abstract: A trajectory planning method for a robotic arm in orchard picking is proposed, utilizing the COA optimization algorithm. The study focuses on a six-joint tandem robotic arm, establishing its mathematical model. Smooth motion trajectories are planned in joint space using the fifth-degree polynomial interpolation method, with constraints on angular displacements, velocities, and accelerations. The COA algorithm is analyzed and optimized, implemented through MATLAB for robotic arm trajectory planning. Experimental results indicate that trajectory planning with the COA algorithm yields a shorter search time for optimal paths compared to traditional methods, and it is more adept at avoiding local optima issues, thus enhancing efficiency in orchard picking with robotic arms.

1. Introduction

In response to the earlier problem of low efficiency of manual work, countries began to use robots to replace manual labor[1], the current robotic arm in the fruit industry orchard picking more and more widely used. With the rapid development of robotics, it has become a hot issue to design high-performance robot arms for orchard picking operations. According to the current robotic arm used in the orchard because of the running path leads to low efficiency problems, this paper is conceived in the robotic arm used in the fruit industry on the basis of the trajectory optimization of the robotic arm in the movement of fruit picking, through the COA optimization algorithm to adjust the running trajectory, so that the trajectory of the movement of the smooth and no mutation, to reduce the impact of the robotic arm to increase the efficiency of the work, as a way to solve the problem[2].

COA algorithm is also known as lobster optimization algorithm, which is a new type of optimization algorithm proposed by the scholar Heming Jia[3] in 2023, COA algorithm which has the ability of searching with high accuracy and not easy to fall into local optimum.

In this paper, COA optimization algorithm is applied to robotic arm trajectory planning to improve the efficiency of robotic arm. Although the COA optimization algorithm has the characteristics of high search accuracy and not easy to fall into the local optimum, but there is the problem of large impact in the movement process of the joints of the robotic arm, in order to reduce the impact of its drawbacks, this paper applies the fifth-degree polynomial algorithm to interpolate the spatial trajectory optimization, so as to make the robotic arm smoother during the movement process, and to reduce the maximum impact.

2. Mathematical modeling of robotic arms

In order to study the trajectory planning of the robotic arm, this paper carries out the modeling of the robotic arm through the robotics toolbox in MATLAB[4]. The modeling of the robotic arm is established by the D-H parameter method, and the D-H parameter table of the robotic arm is shown in Table 1.

Table 1: Robotic Arm D-H Parameter Table.

i	d_i / mm	θ_i (°)	α_i / mm	α_i (°)
1	400	0	25	90
2	0	90	560	0
3	0	0	35	90
4	515	0	0	90
5	0	180	0	90
6	80	0	0	0

In Table 1, i denotes the joints of the robotic arm, θ_i denotes the angle between two axes, z_i and z_{i+1}, in the i-axis direction of the z-axis coordinate system; d_i denotes the distance between two axes, z_i and z_{i+1}, in the i-axis direction of the z-axis coordinate system; a_i indicates the distance between the x_i axis and the x_{i+1} axis in the x_i axis direction; α_i denotes the angle of the distance between the x_i-axis and the x_{i+1}-axis in the direction of the x_i-axis.

The initial position of the robot arm is shown in Fig. 1, and the robot arm base coordinate system is overlapped with the WORLD coordinate system to achieve the function of calculation reduction.

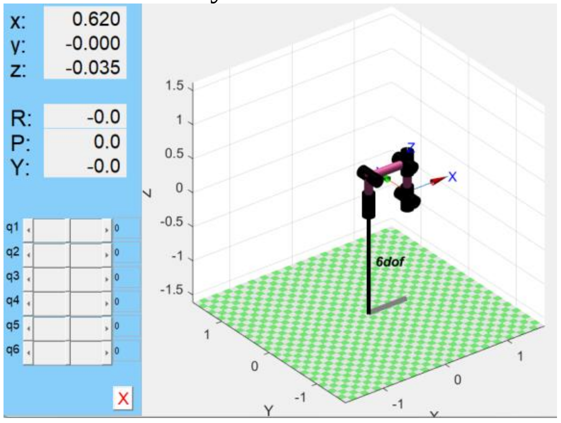
x: 0.620
y: -0.000
z: -0.035
R: -0.0
P: 0.0
Y: -0.0
q1 0
q2 0
q3 0
q4 0
q5 0
q6 0

Figure 1: Robotic arm model.

2.1. Establishment of fifth-degree polynomial interpolation

When the spatial planning of the robotic arm is carried out, the spatial trajectory of the robotic arm can be smoothed by using the method of multinomial interpolation, so that the maximum impact of the robotic arm in the process of movement can be reduced. As for the choice of interpolation, although the low-order interpolation method can make the trajectory curve have a certain degree of smoothness, the acceleration curve will produce discontinuity. The high-order interpolation method is not only too large in calculation, but also prone to produce the Longueuil phenomenon. After comprehensive comparison, this paper uses the fifth-degree polynomial interpolation method[5] to smooth the trajectory of the robotic arm. Its function expression is:

$$\begin{cases} Y(t) = A_5t^5 + A_4t^4 + A_3t^3 + A_2t^2 + A_1t + A_0 \\ \dot{Y}(t) = 5A_5t^4 + 4A_4t^3 + 3A_3t^2 + 2A_2t + A_1 \\ \ddot{Y}(t) = 20A_5t^3 + 12A_4t^3 + 6A_3t^2 + 2A_2 \end{cases} \tag{1}$$

In Equation (1), Y denotes the joint angle at the moment of t, $\dot{Y}$ denotes the joint angular velocity at the moment of t, $\ddot{Y}$ denotes the joint angular acceleration at the moment of t, and $A_0, A_1, A_2, A_3, A_4, A_5$ is the coefficient to be determined. Because of the existence of unknowns, it is necessary to set constraints, defining the start time point t_0 and the end time point t_n, corresponding to the joint angle of γ_0 and γ_n, the initial angular velocity $\dot{\gamma}_0$, the termination angular velocity $\dot{\gamma}_n$, the initial angular acceleration of $\ddot{\gamma}_0$, the termination angular acceleration of $\ddot{\gamma}_n$. Its corresponding information is as in Equation (2):

$$\begin{cases} Y(t_0) = \gamma_0 \\ Y(t_n) = \gamma_n \\ \dot{Y}(t_0) = \dot{\gamma}_0 \\ \dot{Y}(t_n) = \dot{\gamma}_n \\ \ddot{Y}(t_0) = \ddot{\gamma}_0 \\ \ddot{Y}(t_n) = \ddot{\gamma}_n \end{cases} \tag{2}$$

Using the system of fifth-degree polynomial equations as well as the constraints, the six values of the coefficients to be determined are solved for:

$$\begin{cases} A_5 = \dfrac{12(\gamma_n - \gamma_o) - 6(\dot{\gamma}_o + \dot{\gamma}_n)t_n - (\ddot{\gamma}_o - \ddot{\gamma}_n)t_n^{\,2}}{2\mathrm{t}_n^{\,5}} \\ A_4 = \dfrac{30(\gamma_n - \gamma_o) + (16\dot{\gamma}_o + 14\dot{\gamma}_n)t_n + (3\ddot{\gamma}_o - 2\ddot{\gamma}_n)t_n^{\,2}}{2\mathrm{t}_n^{\,4}} \\ A_3 = \dfrac{20(\gamma_n - \gamma_o) - (12\dot{\gamma}_o + 8\dot{\gamma}_n)t_n - (3\ddot{\gamma}_o - \ddot{\gamma}_n)t_n^{\,2}}{2\mathrm{t}_n^{\,3}} \\ A_2 = \dfrac{1}{2}\ddot{\gamma}_o \\ A_1 = \dot{\gamma}_o \\ A_0 = \gamma_0 \end{cases} \tag{3}$$

2.2. Establishing the optimization objective function

In the process of fruit picking by robotic arm, it needs to complete the actions of locating the target, planning the route, grasping the fruits, placing the platform, etc. These actions are more complicated, which will greatly lengthen the working time and reduce the working efficiency. Therefore, this paper optimizes the objective function based on the method of fifth degree polynomial. Here the total working time is taken as the objective function. Let the robotic arm has n segments of trajectory planning, the time of each segment of trajectory planning is, the optimization function is as in equation (4):

$$f(t) = min(\sum_{i=1}^{n} t_i - t_{i-1}) \tag{4}$$

For the mechanical arm in the movement process, easy to produce collision, vibration impact is too large and other instabilities, the use of Monte Carlo method[6], randomly generate 30,000 groups of angle values, the mechanical arm can reach the position of the prediction, as shown in Figure 2, the joint angle, angular velocity, angular acceleration constraints, as a way to reduce the mechanical arm of some of the uncontrollable problems, so that the mechanical arm can be successfully completed the trajectory of the movement.

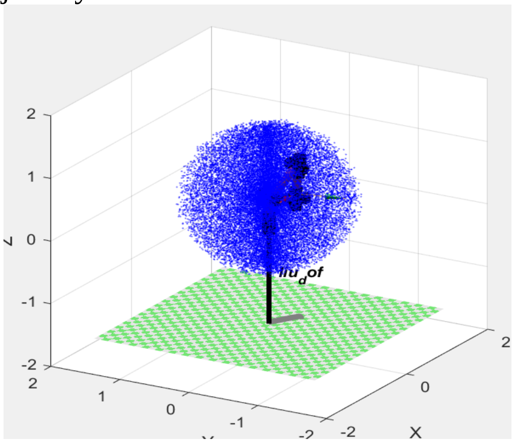

Figure 2: Diagram of the estimated workspace of the robotic arm.

The constraints are:

$$\begin{cases} |\gamma(t)| \leq \gamma(t)_{\max} \\ |\dot{\gamma}(t)| \leq \dot{\gamma}(t)_{\max} \\ |\ddot{\gamma}(t)| \leq \ddot{\gamma}(t)_{\max} \end{cases} \tag{5}$$

3. COA optimization algorithm

The Crayfish optimization Algorithm is an innovative optimization technique inspired by the natural behaviors of crayfish. This algorithm effectively taps into how crayfish search for food, dodge heat, and engage in competition. Essentially, it models the ways crayfish gather together or spread out while foraging and how they react to avoid hot conditions. By replicating these strategies, the Crayfish Algorithm seeks to find the best solutions to complex problems, harnessing the simple, survival-driven actions of these aquatic creatures to tackle challenges in optimization.

3.1. Crayfish stock initialization

In a multidimensional optimization scenario, each shrimp corresponds to a $1 \times m$ matrix, wherein every column of this matrix represents a potential solution to the problem at hand. Within a given set of variables $(X_{i1}, X_{i2}, \ldots, X_{im})$, it is imperative that each variable x_i adheres to predetermined upper and lower bounds. The initialization phase of the Crayfish Optimization Algorithm (COA) entails the stochastic generation of a collection of solution candidates within the space X. These candidate solutions, denoted as X, are meticulously chosen with consideration given to both the population size n and the dimensionality of the problem, denoted as m. The initialization process of the COA algorithm is succinctly encapsulated in equation (6):

$$X = [X_1, X_2, X_3, \ldots, X_n] = \begin{bmatrix} X_{11} & \cdots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \cdots & X_{nm} \end{bmatrix} \tag{6}$$

X is the initialized population position, n is the population number, m is the population dimension, and X_{nm} is the position of individual n in dimension m. The value of X_{nm} is obtained from Eq. (7).

$$X_{nm} = \lambda b_m + (\eta b_m - \lambda b_m) \times rand \tag{7}$$

In Equation (7), λb_m denotes the lower bound of the mth dimension, ηb_m denotes the upper bound of the mth dimension, Rand is the random number.

3.2. Defining Crawfish Temperature and Intake

Influence of temperature on crayfish behavior is pivotal, precipitating distinct stages of activity. Temperature, as delineated in Equation (8), orchestrates these shifts. When temperatures soar beyond 30 ℃, crayfish gravitate towards cooler environs, evading the scorching heat. Within optimal temperature thresholds, crayfish fervently engage in foraging endeavors. Notably, crayfish feeding evinces a pronounced affinity for temperatures hovering around 15 ℃, 25 ℃, and 30 ℃, approximating a normal distribution. Particularly robust foraging behavior is observed within the 20 ℃ to 30 ℃ range. Consequently, the Crayfish Oversight Agency (COA) delineates a temperature span spanning 20 ℃ to 35 ℃. The mathematical manifestation of crayfish feeding, as elucidated in Equation (9), encapsulates these nuanced temperature-dependent nuances.

$$X_{temp} = rand \times 15 + 20 \tag{8}$$

In Eq. (8) X_{temp} denotes the temperature of the environment where the crayfish are located.

$$P = C_1 \times \left(\frac{1}{\sqrt{2 \times \pi \times \partial}} \times \exp\left(-\frac{(X_{temp} - \upsilon)^2}{2\partial^2}\right)\right) \tag{9}$$

Where υ denotes the most suitable temperature for crayfish, ∂ and C_1 denote the intake of crayfish at different temperatures of control.

3.3. Stage of summer vacation

When the temperature exceeds 30 ℃, the temperature is so high that the crayfish will choose to enter a burrow to escape the heat. The burrow x_{burrow} is defined as in equation (10):

$$X_{burrow} = (X_g + X_v)/2 \tag{10}$$

In Eq. (10), X_g denotes the current iterated optimal position, while X_v denotes the optimal position of the current population.

Crawfish competing for burrows is a random thing. When $rand < 0.5$, it means that there are no other crayfish competing for the burrow, and the crayfish will simply enter the burrow to escape the heat. Crayfish enter the burrow for summer vacation as follows in equation (11):

$$X_{nm}^{j+1} = X_{nm}^{j} + C_2 \times rand \times (X_{burrow} - X_{nm}^{j}) \tag{11}$$

In Eq. (11), j denotes the number of current iterations and $j+1$ denotes the number of next-generation iterations. C_2 is the decreasing curve, as shown in Eq. (12).

$$C_2 = 2 - (j/J) \tag{12}$$

J denotes the maximum number of iterations.

In the summering phase, crayfish aim to enter the burrow for summering, which represents the optimal solution. This brings the individual closer to the optimal solution and enhances the development of the COA. Allowing the algorithm to converge faster.

3.4. Competition stage

When $X_{temp} \geq 30$ and $rand \geq 0.5$, it means that there are other crayfish interested in the burrow. At this point, the crayfish will start to compete for the burrow. As shown in equation (13):

$$X_{nm}^{j+1} = X_{nm}^{j} - X_{lm}^{j} + X_{burrow} \tag{13}$$

In Equation (13), l denotes a random individual of crayfish as shown in Equation (14):

$$l = round(rand \times (n-1)) + 1 \tag{14}$$

In the competition phase, the crayfish compete with each other for burrows, and the crayfish X_i adjusts its position according to the X_l position of another crayfish. By adjusting the position, the search range of the algorithm is enlarged and the exploration ability of the algorithm is enhanced.

3.5. Foraging stage

When $X_{temp} \leq 30$, the temperature is suitable for crayfish foraging. The crayfish will move towards the food. After finding the food, the crayfish will judge the size of the food. If the food is too large, the crayfish will tear the food with its claws and eat it alternately with its second and third walking feet. The food location X_{food} is defined:

$$X_{food} = X_g \tag{15}$$

The food size Q is defined as:

$$Q = C_3 \times rand \times (fit_i / fit_{food}) \tag{16}$$

In Eq. (16), C3 is the food factor, which represents the largest food with a value of constant 3, fit_i represents the adaptation value of the ith crayfish, and fit_{food} represents the adaptation value of the food location.

The crayfish's judgment of food size comes from the size of the largest food. When $Q > (C_3 + 1)/2$, it means the food is too big. At this point, the crayfish will tear the food with its first clawed foot. The mathematical equation is as follows:

$$X_{food} = \exp(-\frac{1}{Q}) \times X_{food} \tag{17}$$

When the food is torn and reduced in size, the second and third paws pick up the food and put it in the mouth. In order to simulate the alternating process, a combination of sine and cosine functions were used to simulate the alternating process. The food obtained by the crayfish is also related to the amount of food intake, and the equation for foraging is as follows:

$$X_{nm}^{j+1} = X_{nm}^{j} + X_{food} \times P \times (\cos(2 \times \pi \times rand) - \sin(2 \times \pi \times rand)) \tag{18}$$

When $Q \le (C3+1)/2$, the crayfish moves toward the food and feeds directly with the following equation:

$$X_{nm}^{j+1} = \left(X_{nm}^{j} - X_{food}\right) \times P + P \times rand \times X_{nm}^{j} \tag{19}$$

During the foraging phase, crayfish use different feeding methods depending on the size of their food Q. The food X_{food} represents the optimal solution. When the size Q of the food is suitable for the crayfish to eat, the crayfish will approach the food.

When Q is too large, it indicates a significant difference between the crayfish and the optimal solution. Therefore, X_{food} should be reduced and made closer to food. And controlling the crayfish ingestion increases the randomness of the algorithm. The foraging phase enables the COA algorithm to approach to the optimal solution, which enhances the development of the algorithm and makes the algorithm have good convergence ability.

3.6. COA Algorithm Flow

Parameter Definition and Population Initialization Define the number of iterations j, the population n, the dimension m, the upper bound ub and the lower bound lb. Initialize the population X according to the upper and lower bounds. The initialization of the population is obtained from the equation. Temperature Definition The ambient temperature of crayfish is defined by Eq. to bring the COA into different stages of development. The summering phase and the competition phase. When the temperature is greater than 30 ° and $rand$ < 0.5, the COA enters the summering stage. At this point, the COA gets a new position based on the cave position and crayfish position. When the temperature was less than 30 ° and $rand$ ≥ 0.5, COA entered the competition phase. At this time, two crayfish will compete for the burrow. And the new position is obtained according to the burrow position and location of the two crayfish. The evaluation function evaluates the overall and determines whether to exit the cycle. The most suitable adaptation value is output. As shown in Figure 3.

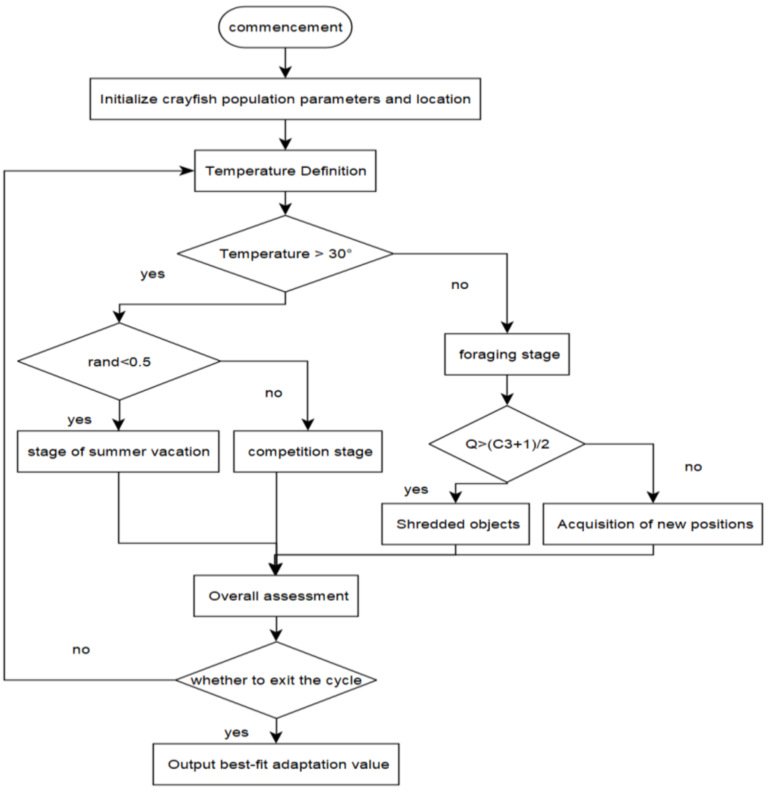

Figure 3: COA Algorithm Flowchart.

4. Experimental Simulation and Analysis

In this paper, we study and analyze trajectory planning with a six-degree-of-freedom robotic arm and use robot tool box in MATLAB software for analytical modelling[7]. Here in this paper, the starting point coordinate A(2.4, -0.7, 2.4) and the end point coordinate B(0.32, 0.76, 1.28) are set to plan the trajectory of the robotic arm, and according to the COA optimization algorithm, the trajectory graph is obtained as shown in Figure 4.

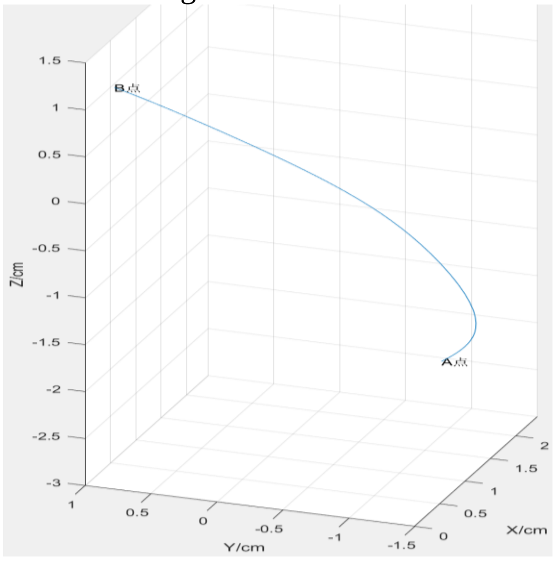

Figure 4: Robotic arm trajectory diagram.

In order to verify the feasibility and optimization effect of COA-based optimization algorithm applied in robotic arm trajectory planning, this paper introduces the traditional particle swarm algorithm and whale algorithm for comparison, as shown in Figure 5. Here, the parameter values of each algorithm are tuned to be consistent: the population size is 100 and the number of iterations is 50. Through Figure 5, it can be clearly seen that from the beginning of the COA optimization algorithm is obviously faster than the other two algorithms convergence speed, the number of iterations in the 20 to 50 times, the COA optimization algorithm convergence accuracy is higher and more stable. This confirms the feasibility of COA optimization algorithm applied to robotic arm trajectory planning, and compared with the other two algorithms, COA optimization algorithm has a higher search accuracy and a lower possibility of falling into the local optimum.

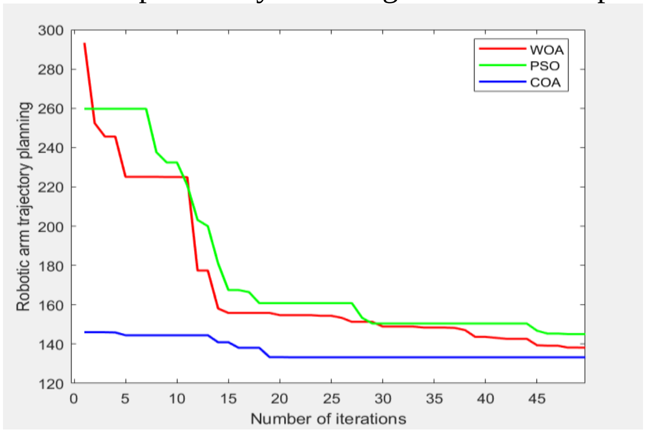

Figure 5: Fitness curve.

The above COA optimization algorithm is applied to the robotic arm trajectory planning, and the spatial trajectories of the robotic arm joints are smoothed by a fifth-degree polynomial, and the curves of the angular displacements, angular velocities and angular accelerations of the six joints are obtained by applying the MATLAB simulation as shown in Figs. 6, 7 and 8.

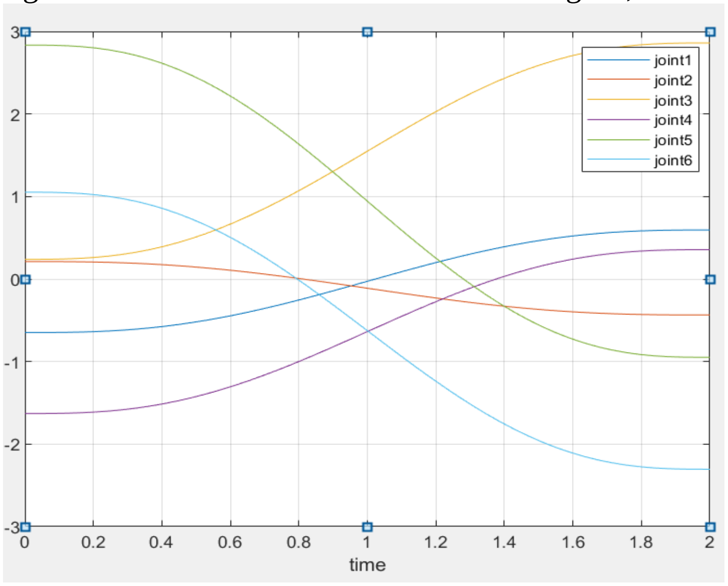

Figure 6: Angular displacement graph.

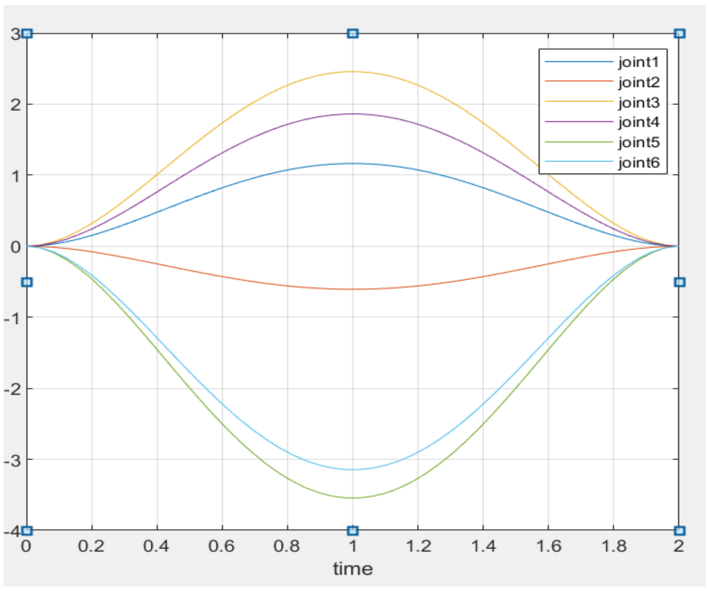

Figure 7: Angular velocity graph.

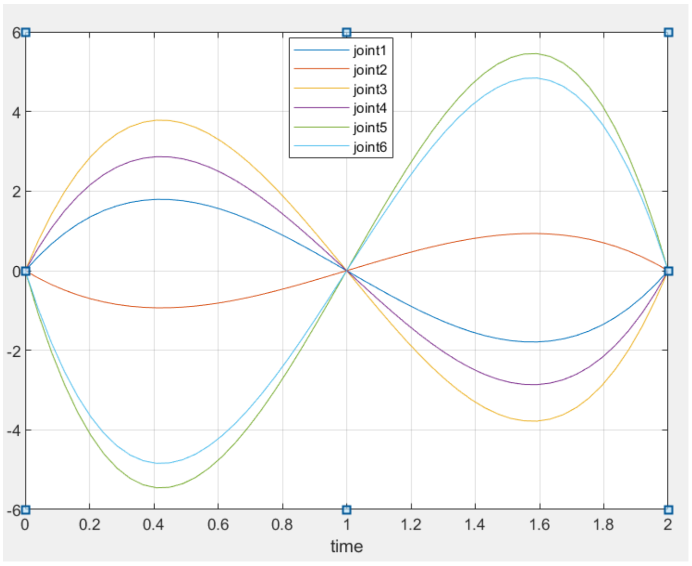

Figure 8: Angular acceleration graph

As can be seen in Figures 6, 7 and 8, the curves of angular displacement, angular velocity and angular acceleration of the COA optimization algorithm using fifth degree polynomials are obviously continuous, smooth and free of sudden changes, which indicates that the robotic arm of this paper has a good stability for trajectory planning movement in the joint space. And this paper to prevent the mechanical arm movement to produce collision, vibration impact is too large and other uncontrollable problems, add the equation 5 as a constraint, so that the mechanical arm can be safe and stable movement, to verify the feasibility and rigor of the algorithm proposed in this paper[8].

5. Conclusion

In this paper, a six-joint tandem robotic arm is taken as an experimental object, its mathematical model is established, and a smooth motion trajectory is planned in the joint space by using the fifth-degree polynomial interpolation method to constrain the angular displacements[9], angular velocities, and angular accelerations of the joints of the robotic arm, so as to ensure the continuity and safety of the robotic arm in the motion process. The idea of applying the COA algorithm in the trajectory planning of the robotic arm is proposed, and its feasibility is verified by simulation. The application of COA algorithm in robotic arm trajectory planning is realized through MATLAB. Comparing the COA algorithm with the traditional particle swarm and whale algorithms, the trajectory planning of the COA algorithm has a shorter time of searching the optimal path than the traditional algorithm[10], and it is easier to jump out of the local optimal problem. This algorithm can effectively improve the stability of the robotic arm and provide help for the robotic arm in orchard picking.

References

[1] Savsani P, Jhala R L, Savsani V J. Comparative study of different metaheuristics for the trajectory planning of a robotic arm[J]. IEEE Systems Journal, 2014, 10(2): 697-708.

[2] Hao W G, Leck Y Y, Hun L C. 6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control[C]//2011 4th international conference on mechatronics (ICOM). IEEE, 2011: 1-7.

[3] Jia H, Rao H, Wen C, et al. Crayfish optimization algorithm[J]. Artificial Intelligence Review, 2023, 56(Suppl 2): 1919-1979.

[4] Gasparetto A, Boscariol P, Lanzutti A, et al. Path planning and trajectory planning algorithms: A general overview [J]. Motion and Operation Planning of Robotic Systems: Background and Practical Approaches, 2015: 3-27.

[5] Shin K, McKay N. A dynamic programming approach to trajectory planning of robotic manipulators [J]. IEEE Transactions on Automatic Control, 1986, 31(6): 491-500.

[6] Rajan V. Minimum time trajectory planning[C]//Proceedings. 1985 IEEE International Conference on Robotics and Automation. IEEE, 1985, 2: 759-764.

[7] Gasparetto A, Zanotto V. Optimal trajectory planning for industrial robots [J]. Advances in Engineering Software, 2010, 41(4): 548-556.

[8] Roy R, Mahadevappa M, Kumar C S. Trajectory path planning of EEG controlled robotic arm using GA [J]. Procedia Computer Science, 2016, 84: 147-151.

[9] Dubowsky S, Norris M, Shiller Z. Time optimal trajectory planning for robotic manipulators with obstacle avoidance: a CAD approach[C]//Proceedings. 1986 IEEE International Conference on Robotics and Automation. IEEE, 1986, 3: 1906-1912.

[10] Tian L, Collins C. An effective robot trajectory planning method using a genetic algorithm[J]. Mechatronics, 2004, 14(5): 455-470.