Research on path and energy consumption control of six-axis manipulator based on multi-objective optimization method

Zhongyuan Ma, Anbang Zhu, Haidong Feng, Chenqi Zhou, Hao Liu, Dongping Sheng, Hun Guo, Chun Su*

Changzhou Institute of Technology, Changzhou, China

ABSTRACT

In the present rapidly evolving industrial and technological domains, robotic arms have emerged as crucial automated equipment for enhancing production efficiency, accuracy, and safety. The six-degree-of-freedom robotic arm has been extensively utilized in multiple fields like industrial production, precise operation, operations in hazardous environments, and logistics on account of its capacity to undertake complex tasks. Nevertheless, in complex circumstances, the path optimization and joint angle optimization of robots have consistently been significant factors influencing their working efficiency. This paper intends to optimize the path and joint angles of the six-axis robotic arm through establishing a multi-objective optimization model, with the aim of improving end accuracy and reducing energy consumption.

For work one, a mathematical model for the robotic arm was crafted using the Denavit-Hartenberg (D-H) parameter method, enabling the construction of a homogeneous transformation matrix and the establishment of a forward kinematics model. This foundation allowed for the formulation of a nonlinear optimization model, which, coupled with the particle swarm optimization (PSO) algorithm, was used to minimize the end effector's positional error, achieving a significant reduction to 1.0×10^-3 mm and thus enhancing the arm's precision. For work two, expanding upon work one, an energy consumption model was developed to account for both kinetic and potential energies, ensuring that the terminal error remained within an acceptable range. A multi-objective optimization model was formulated with the goal of minimizing both terminal error and energy consumption. The PSO algorithm was employed once more, yielding an optimal joint angle configuration that resulted in minimal energy consumption of 23.9261 J and a terminal error of 3.044×10^-8 mm. For work Three, In addressing obstacle navigation, a raster map was created, and the A* algorithm was utilized to plan the optimal movement path for the robotic arm's base. This approach, built upon the previously established model, determined an optimal joint angle configuration that effectively circumvents obstacles while maintaining high operational accuracy and energy efficiency, with a terminal error of 4.028×10^-7 mm. For work four, the work of grasping multiple target points amidst obstacles was modeled as a Traveling Salesman work (TSP). A genetic algorithm was applied to this model to determine the optimal path for sequential visits to target points and calculate the joint angles, terminal error values, and energy consumption for each point. This resulted in a maximum terminal error within an acceptable range and an optimized energy consumption of 111.9223 J.

This study significantly advances the operational efficiency and accuracy of robotic arms in complex environments through robust mathematical modeling and optimization algorithms. It contributes to the evolution of robotic arm technology towards greater intelligence, flexibility, and efficiency, setting a foundation for broader applications in future industrial and technological advancements.

Keywords: D-H Parameter Method, Genetic Algorithm, Particle Swarm Optimization, Multi-Objective Optimization, A* Algorithm, TSP

1 INTRODUCTION

In the contemporary rapidly advancing industrial and technological fields, robotic arms have become key automated equipment for enhancing production efficiency, precision and safety. The application of six-degree-of-freedom robotic arms in industrial automation is increasingly widespread, and their precise control is crucial for improving production efficiency and product quality. Dynamics parameter identification, as the basis of robotic arm control, holds significant importance for achieving precise motion control. Ye Teng (2023) thoroughly explored the issue of dynamics parameter identification of six-degree-of-freedom humanoid robotic arms in his research and proposed effective trajectory planning methods, providing a theoretical basis and technical support for the precise control of robotic arms [1]..

Research on robotic arms primarily focuses on kinematic and dynamic modeling, optimal design of joint angle paths, and path planning. The overarching aim of these studies was to enhance the performance and applicability of robotic arms, ensuring that they operate with both efficiency and accuracy. Among these areas, optimizing the joint angle path is particularly crucial, as it directly influences the operational precision and energy consumption of the robotic arm. Optimal design must consider multiple, often conflicting, objectives, such as minimizing output and terminal errors, while also reducing energy consumption within acceptable error limits.

In practical applications, robotic arms must accurately reach the target position while maintaining their energy efficiency. For instance, when picking up objects, the end effector of the robotic arm does not need to perfectly align with the target's center if the energy consumption can be reduced by optimizing the joint angle path, thus allowing for a small margin of error. Additionally, when performing tasks such as grabbing items at various locations along a production line, the robotic arm must balance the terminal error and energy consumption, necessitating a comprehensive optimization of both the base movement and joint angle paths.

In industrial production settings, the efficient operation of robotic arms can significantly enhance production output and product quality. For example, in the plastics manufacturing industry, the application of robots has been proven to increase production efficiency and product quality, as thoroughly explored in the research by Shi, L. C. [2]. Similarly, in the electronics manufacturing process, robotic arms can precisely install solder components, ensuring high product accuracy and reliability.

To achieve these applications, robotic-arm path planning and joint angle optimization require advanced algorithms and technologies. Techniques such as genetic algorithms, particle swarm optimization, and A* algorithms are effective for solving the path-planning challenges of robotic arms in complex environments, ensuring obstacle avoidance and optimal pathfinding. In addition, integrating deep learning and reinforcement learning technologies enables robotic arms to autonomously learn and adapt to dynamic environments, thereby enhancing their intelligence and operational capabilities.

For work one: A simplified diagram of the six-degree-of-freedom robotic arm in the zero position was drawn, and a kinematic model was established based on the provided initial parameters. To enable the robotic arm to perform the grabbing task, the target point was positioned at distances of (1500, 1200, and 200 mm) from the robotic arm. The joint angle path was then optimized to minimize the end error[3].

For work two: Building on work 1, the total mass of the robotic arm was 5 kg. The optimization process considers the moment of inertia and the average angular velocity of the joints. With an allowable end error of ± 200 mm, the joint angle path was optimized to minimize both end error and energy consumption. This approach ensures that the robotic arm operates efficiently while maintaining the accuracy within a specified error range.

For work three: Based on work two, the robotic arm must bypass obstacles to grab the goods. The base was treated as a mass point, with energy consumption considered only during the grasping process. The base should be moved near the target point, the goods are grabbed, and the base returns to the starting point. Optimal base movements and joint angle paths were designed, and the path was visualized on a raster map to avoid obstacles.

For work four: The robotic arm performs multi-target point-grabbing tasks and bypasses obstacles. Optimal base movement and joint angle paths are designed and visualized on a raster plot. The total terminal error and energy consumption were specified in the results to ensure that both were minimized.

2 RELATED WORK AND ASSUMPTIONS

2.1 Work analysis

For work one, a mathematical model of a robotic arm with six degrees of freedom was established. The kinematic model was developed with reference to the structural design analysis of a robotic arm with six degrees of freedom by Pan Mingzhang et al. (2024) to ensure the accuracy and practicality of the model [4].The joint angle path is then optimized to minimize the error of the end effector. During the construction of the kinematic model and the optimization objective function, the research findings of Chen Yonggang (2022) were referenced, particularly his in-depth analysis of trajectory planning, which guided the determination of key parameters and the optimization direction of the model[5]. In the zero-position state, the initial DH parameters and the target point position of the robotic arm are provided. A mathematical model describing the motion of the robotic arm was constructed using the DH parameter method[6]. A nonlinear optimization algorithm was applied to adjust the joint angles of the robotic arm, ensuring that the end effector approached the target point as closely as possible. The optimization results demonstrate that the terminal error is effectively controlled within a minimal range, thereby significantly enhancing the positioning accuracy and operational efficiency of the robotic arm, which aligns with the findings of high-precision control strategies discussed by Wang, Q. F. [7].

In work two, the research focused on minimizing the movement energy consumption of the robotic arm while ensuring that the end error remained within the allowable range. Considering the mass of the robotic arm, moment of inertia, and average angular velocity of the joints, an energy consumption model was established that comprehensively accounts for both kinetic and potential energy. The particle swarm optimization algorithm was employed to optimize the joint angle path to minimize energy consumption [8]. The optimization results indicate that while satisfying the terminal error requirements, the energy consumption of the robotic arm was significantly reduced. This reduction is crucial for enhancing the operational efficiency of the robotic arm and lowering operating costs.

In work three, a more complex scenario was addressed, involving the planning of both the base movement path and the joint angle path of the robotic arm in the presence of obstacles. Various algorithms and techniques have been proposed to enhance performance under such conditions. Notably, research by Guo Mengshi et al. (2018) offers a path planning method that shows promise in ensuring path smoothness and reducing motion errors in robotic arms [9]. This study applies the A* algorithm to plan the movement path of the base [10] and integrates it with a genetic algorithm to optimize joint angles with the aim of avoiding obstacles while minimizing terminal errors and energy consumption. This approach enables the robotic arm to effectively navigate around obstacles and reduces energy consumption while maintaining operational accuracy.

In work four, the task involves solving the path optimization work for a robotic arm performing multitarget point grabbing tasks in a complex environment. This work is modeled as a Traveling Salesman work (TSP), and a genetic algorithm is employed to find the solution, with the primary objective of minimizing both the total terminal error and total energy consumption. The research findings indicate that the robotic arm can sequentially visit all target points and return to the starting point along the optimal path while satisfying the optimization criteria for the terminal error and energy consumption. This study has significant theoretical and practical value for enhancing the efficiency and accuracy of robotic arms in real-world applications.

2.2 Model assumption

1) Joint motion assumption: The motion of the robotic arm joints adheres to the rigid link theory, where the length of the lever between each joint remains constant. The angular velocities of the joints were uniformly distributed, and the impact of the acceleration changes on the motion path was not considered.

2) Energy consumption assumption: The primary energy consumption of the robotic arm is attributed to joint rotation and potential energy work against gravity. Other factors such as motor efficiency and friction were not considered. The combined mass of the robotic arm and the end load was 5 kg. The moment of inertia and average angular velocity for each joint are listed in Table 2.

3) Obstacle assumption: During movement, the base of the robotic arm is treated as a particle and its energy consumption is not considered. The base cannot move diagonally, and all the joints must avoid passing over obstacles.

4) Dynamics Assumption: The robotic arm is assumed to remain in a stable state throughout task execution, with no deviations caused by vibrations or other dynamic factors. The acceleration due to gravity, g, was considered a constant value of 9.81 m/s^2 .

5) Path Planning Assumption: In the path-planning process, both the base and joints of the manipulator can move simultaneously, but they are handled separately in calculations to simplify the work. The optimization algorithm is assumed to converge to the global optimal

solution or an approximate optimal solution without being affected by external interference during the calculation process.

6) Environmental Assumption: The robotic arm operates in a two-dimensional plane, where the positions of the base and obstacles are fixed. The dimensions and locations of the cargo and obstacles were known and remained constant throughout the operation of the robotic arm.

3 MODEL BUILDING AND ANALYZING

3.1 Model establishing and analyzing for work one

3.1.1 Model for Transportation of Personnel and Supplies

The D-H parameter method (Denavit-Hartenberg parameter method) is a method proposed by Denavit and Hartenberg to describe the motion relationship of multi-joint mechanisms such as robotic arms through link parameters [11]. By establishing a link coordinate system, four link parameters can be used to describe the geometric size and connection relationship of each link in the robotic arm. The specific steps are as follows.

Steps to establish the connecting rod coordinate system

1. Find each joint axis as follows:

The axis direction along the axis of the i+1 joint.

2. Determine the origin of the coordinate system.

If joint i and joint axis $i+1$ are in different planes, draw a common perpendicular line between joint axis i and $i+1$, and use the point where the common perpendicular line intersects joint axis i as the origin of coordinate system {i}. If the joint axes i and $i+1$ are on the same plane, then the point where the two joint axes intersect is the origin of the coordinate system {i}.

If joint axes i and $i+1$ are on the same plane, the point at which the two joint axes intersect is the origin of coordinate system {i}.

3. Determine the direction of the coordinate axis.

If axes i and $i+1$ do not intersect, it is specified that axis x_i is along the common perpendicular direction of axis i and axis $i+1$, and the direction away from axis i is the positive direction of the x_i axis; if axis i and axis $i+1$ intersect, axis x_i is perpendicular to the intersection surface of the two axes.

The direction of the coordinate axis y_i was determined according to the right-hand rule. Description of the four connecting rod parameters. After the link coordinate system is established, four link parameters can be used to describe the geometric dimensions of each link of the robotic arm.

1. Connecting rod length: the distance between the axes of two adjacent joints measured along x_i.

2. Connecting rod torsion angle: represents the angle between the axes of the two adjacent joints rotating along the x_i-axis.

3. Connecting rod distance: This is the rotation axis deviation between adjacent connecting rods measured along the z-axis.

4. Connecting rod rotation angle: This is the angle between the two common vertical lines and the z_i axis, rotating along the z_i axis.

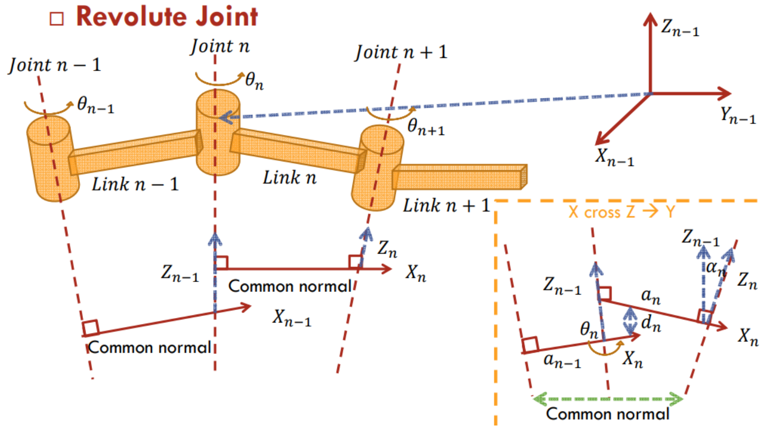

Figure 1 D-H parameter method to establish coordinate system

A simplified diagram of the six-degree-of-freedom manipulator that establishes the zero position through the initial D-H parameters of the manipulator based on the data in Table 1 is shown in Figure 2.

Table 1 Initial DenDenavit-Hartenberg (D-H) parameters of the robotic arm

Joint i	a_{i-1}/(mm)	α_{i-1}/(°)	d_i/(mm)	θ_i Range of joint variation/(°)
1	0	0	600	$160\sim-160$
2	300	-90	0	$-150\sim15$
3	1200	0	0	$-200\sim80$
4	300	-90	1200	$-180\sim180$
5	0	-90	0	$-120\sim120$
6	0	-90	0	$-180\sim180$

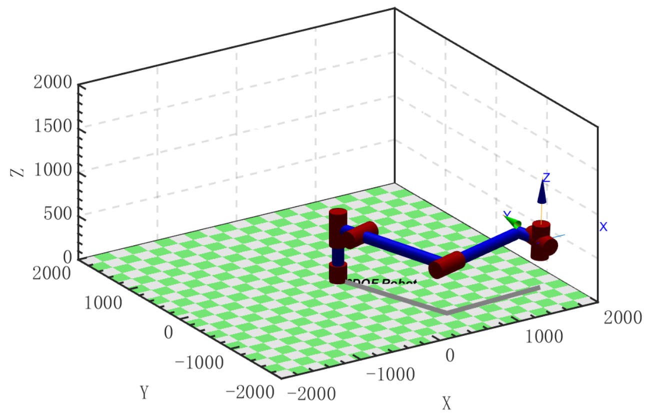

Figure 2 simplified diagram of the robotic arm

3.1.2 Forward and inverse kinematics analysis

(1) Forward kinematics

In this study, a method similar to that of Chen et al. (2023) was used to establish a kinematic model of the robotic arm. Building on this model, trajectory planning was optimized to achieve more efficient energy use and precise control of the end effector [12]. Forward kinematics involves determining the pose of the end effector of a robotic arm based on known joint angles. The homogeneous transformation matrix for the adjacent links of the robotic arm is established using the improved D-H parameter method, and these matrices are multiplied to obtain the pose of the end effector relative to the base coordinates.

For the i_{th} joint, the transformation matrix is as follows:

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\cos\alpha_i & \alpha_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\cos\alpha_i & \alpha_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

Multiply the inverse of the rotation matrix of the first joint by the left side of the pose matrix of the robotic arm:

$$T_0^6 = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \quad (2)$$

(2) Inverse kinematics

The inverse kinematics of the robotic arm are fundamental to motion control and play a crucial role in the trajectory planning of the robotic arm. Inverse kinematics involve calculating each joint variable based on the known target posture of the end effector. Various methods exist for solving the inverse kinematics work, including analytical and numerical approaches. In this case, the work was solved numerically. The process for solving the inverse kinematics of the robotic arm is shown in Figure3.

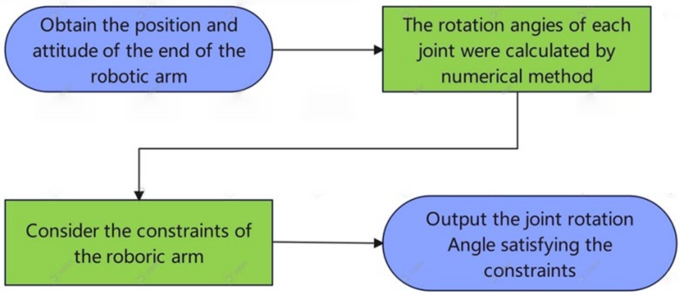

Figure 3 inverse kinematics solution process

The robot arm pose matrix on the left is multiplied by the inverse of the first joint rotation matrix:

$$ {}^0_1A^{-1} \cdot {}^0_6T = {}^2_1A \cdot {}^2_3A \cdot {}^3_4A \cdot {}^4_5A \cdot {}^5_6A = {}^1_6A \quad (3)$$

A system of equations was established by equating the corresponding elements in the transformation matrix, enabling the calculation of the angular displacement for each joint. Given the complexity of the kinematics of the robotic arm, multiple solutions may arise, requiring additional analysis to identify the optimal solution.

3.1.3 Optimizing objective function

The optimization objective function aims to minimize the Euclidean distance between the position of the end effector and the target position, thus defining the error as:

$$E = \sqrt{(X_{target} - X_{end})^2 + (Y_{target} - Y_{end})^2 + (Z_{target} - Z_{target})^2} \quad (4)$$

Optimization goals:

$$MinE(\theta) = \sqrt{(x(\theta) - 1500)^2 + (y(\theta) - 1200)^2 + z(\theta) - 200)^2} \quad (5)$$

Where, $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$.

To address the path optimization work of a six-degree-of-freedom manipulator, a nonlinear optimization algorithm capable of handling complex multivariable optimization challenges is employed. By applying this algorithm [13], the optimized joint angles and the corresponding end-effector positions were determined, as shown in Table 2, Table 3, and Figure 4.

Table 2 Optimized joint angles

Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
78.0012	-78.0011	4.7814	84.4210	0	0

Table 3 optimized end effector position

X/mm	Y/mm	Z/mm
1500	1200	200

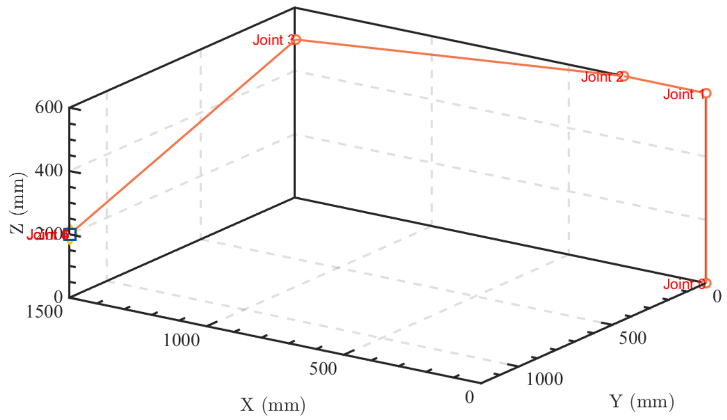

Figure 4 optimized six-degree-of-freedom manipulator

The data in the table and Figure 4 demonstrate that the optimized joint angles satisfy the constraints specified in the work, and the position of the optimized end effector aligns with the target position, with an error as small as 1.0×10^-3 mm.

Through these steps, a kinematic model of the six-degree-of-freedom robotic arm is established, and a joint angle path optimization method is proposed to minimize the final error. This framework lays a solid foundation for further research and application.

3.2 Model establishing and analyzing for work one

3.2.1 Model establishment

(1) Terminal error model:

End-position calculation: The position of the robot arm can be calculated according to the joint angle and kinematic model of the robot arm. The forward kinematic formula can be used to obtain the end coordinates (x_t, y_t, z_t).

Error definition: Defining the target point as, terminal error E can be expressed as

$$(x_t, y_t, z_t) = \sqrt{(x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2} \tag{6}$$

Error limit: Ensure.$E \leq 200mm$

(2) Energy-consumption model

The energy consumption was calculated using the joint energy consumption rotation parameters in Table 2, the joint's rotational inertia III and angular velocity, and the change in gravitational potential energy. The calculation formula is as follows:

$$P_i = \frac{1}{2} I_i w_i^2 \tag{7}$$

$$\Delta E_{potential} = mg\Delta h \tag{8}$$

where m represents the total weight of the robotic arm (5 kg), is the acceleration due to gravity, and h is the height variable.

Thus, the total energy consumption is:

$$P_{total} = \sum_{i=1}^{6} \frac{1}{2} I_i w_i^2 + mg\Delta h \tag{9}$$

where is the moment of inertia of the i_{th} joint and is the average angular velocity of thei_{th} joint.

Among these parameters, I_i represents the moment of inertia of the i_{th} joint, and $w_i$is the average angular velocity of the i_{th} joint.

3.2.2 Optimization function

The optimization goal was to minimize the terminal error, E, and energy consumption. The objective function is defined as:

$$F(\theta) = \alpha E + \beta P_{tatal} \tag{10}$$

Among them,α and$\beta$are the weight coefficients, and$\alpha + \beta = 1$.

Constraints:

$$s.t = \begin{cases} E \leq 200 \\ \theta_i^{min} \leq \theta_I \leq \theta_i^{max} \end{cases} \tag{11}$$

The above model is a multi-objective optimization model, and the particle swarm optimization algorithm is employed to solve it. The specific solution steps are as follows:

Step one: Initialize the particle swarm and randomly generate a set of joint angles as the initial solution.

Step two: For each particle, calculate its objective function value and constraints.

Step three: Iteratively update individuals or particles, gradually approaching the optimal solution.

The particle swarm algorithm is used to determine the minimum energy consumption and the optimal joint angles, as shown in Figure 5 and Table 4.

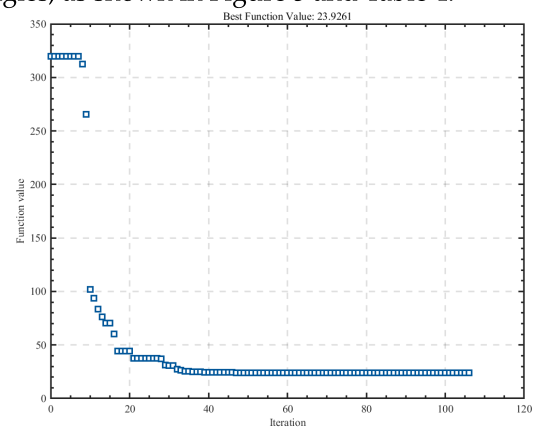

Figure 5 minimum energy consumption

Table 4 joint angles of each joint after optimization

Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6

-15	15	32.0883	-84.4208	-73.2876	1.83856

Analysis of Figure 5 and Table 4 reveals that the minimum energy consumption is 23.9261 J, and the minimum terminal error is 3.044×10^-8 mm. This error is extremely small and essentially meets the requirements of the work.

3.3 Model establishing and analyzing for work three

3.3.1 Model definition

In this scenario, the robotic arm must bypass obstacles to grab the goods, with the base moving near the target point before executing the grabbing action. The objective is to minimize terminal errors and energy consumption while considering the impact of obstacles.

As indicated in the work statement, the initial state of the robotic arm is the zero position given in work 1. To simplify the work, only the energy consumption during the robotic arm's grasping process is considered, excluding the energy consumption related to movement.

(1) Variable definition

Base position: $P_{base} = (x_{base}, y_{base})$:The position of the base in the grid.

Target location: $P_{target} = (x_{target}, y_{target})$: The position of the target in the grid.

Obstacle location: $P_{obstacle} = (x_i, y_i)$: The position of the obstacle in the grid.

End effector position: $P_{end} = (x_{end}, y_{end})$::: Can be solved by inverse kinematics

Joint angle: θ_i(i=1，2，3，4，5，6): Angle of thei_{th} joint.

(2) Establishment of objective function:

Establish the same objective function as in work 2:

$$F(\theta) = \alpha e + \beta P_{tatal} \tag{12}$$

Among these, E represents the terminal error, and C denotes the energy consumption.

Constraints:

1）Movement Constraints: The base must avoid all obstacles during movement.

2）Grasp Constraint: The end effector must be positioned above the target.

3）Joint Angle Constraints: Each joint angle must remain within its specified range of motion.

3.3.2 Path planning

Step one: Mark the locations of the obstacles, the base, and the target location as provided in the work's attachment on a grid diagram. This is illustrated in Figure 6.

$$G_{ij} = \begin{cases} 0 & Clear\ of\ obstacles \\ 1 & There\ is\ an\ obstacle \end{cases} \tag{13}$$

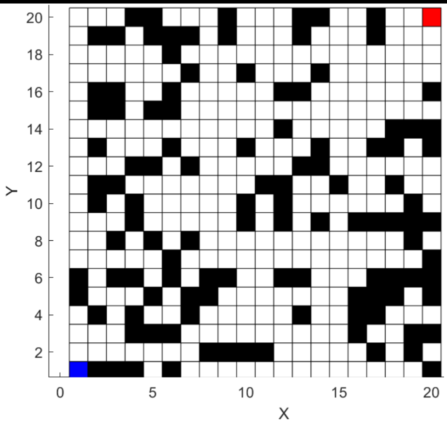

Figure 6 Raster diagram

In the Figure 6, blue represents the starting point, black indicates obstacles, and red marks the target point.

Step two: Base Path Optimization The A* algorithm is employed to determine the path from the starting point to the target location. The steps of the A* algorithm are illustrated in Figure 7 below:

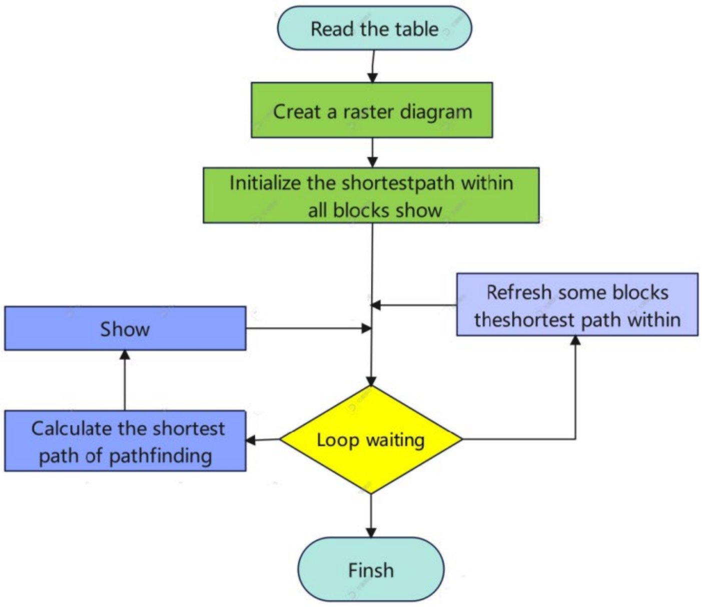

Figure 7 A algorithm flow chart*

In computer science, the A* algorithm, as an extension of the Dijkstra algorithm, is widely used in pathfinding and graph traversal because of its efficiency. It is widely used in games such as StarCraft.

These include: The Search Area: The search area in the picture is divided into a simple two-dimensional array. Each element of the array corresponds to a small square. Of course, who can also divide the area into five-pointed stars, rectangles, etc., usually a The center point of the unit is called the search area node.

Open List: Storing the nodes to be detected in the path planning process in the Open List, and the grids that have been detected are stored in the Close List.

Parent node: The node used for backtracking in path planning. It can be considered as the parent node pointer in the doubly linked list structure during development.

Path Sorting: The specific node to move to is determined by the following formula: F(n) = G + H. G represents the movement cost from the initial position A along the generated path to the specified grid to be detected. H specifies the estimated movement cost of the grid to be tested to the target node B.

Heuristics Function: H is a heuristic function, which is also considered a heuristic. Since not sure what obstacles will appear in front of us before finding the only path, that can use an algorithm to calculate H. According to Depends on the actual scenario. In our simplified model, H uses the traditional Manhattan Distance, which is the sum of horizontal and vertical distances.

In order to facilitate the visualization of the results, the content in the attachment is reset and changed to the coordinates in the normal rectangular coordinate system. The optimal path from the starting point to the target point is solved through the A* algorithm, as shown in Figure 7:

In computer science, the A* algorithm, an extension of the Dijkstra algorithm, is widely used in pathfinding and graph traversal due to its efficiency. It is particularly popular in games like StarCraft. The key components of the A* algorithm include:

Search Area: The search area is divided into a simple two-dimensional array, with each element corresponding to a small square. The area can also be divided into other shapes like stars or rectangles. Typically, the center point of each unit is referred to as a search area node.

Open List: During path planning, nodes that need to be explored are stored in the Open List, while those that have already been explored are stored in the Close List.

To visualize the results more clearly, the content from the attachment has been reset and adapted to a standard rectangular coordinate system. The optimal path from the starting point to the target point is determined using the A* algorithm, as shown in Figure 8.

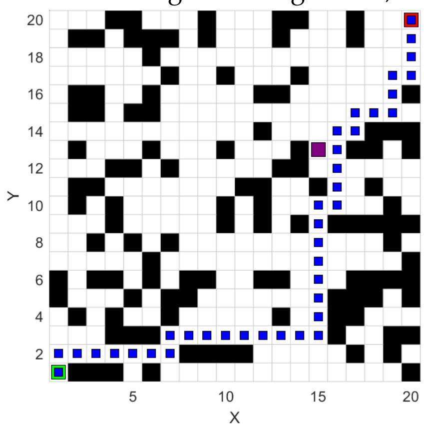

Figure 8 A algorithm path optimization*

As shown in Figure 8, the optimal location for the robotic arm base, calculated using the A* algorithm, is at coordinates (20, 19). The optimal path is highlighted in the purple area on the figure.

3.3.3 Results evaluation

Path Validity: When the robotic arm base is positioned at (20, 19), as indicated in the work, each grid measures 200×200 mm, and the cargo height is 200 mm. Assuming the robotic arm base is always centered within its grid, the cargo's coordinates relative to the robotic arm base would be (0, 200, 200). According to the information provided in work 1, each joint of the robotic arm has a specific length. If the base remains at (20, 19), it fails to meet the requirements for picking up the goods. This positioning creates a dead zone where the robotic arm is unable to reach the cargo, making it impossible to complete the task. Consequently, it is necessary to adjust the optimal stopping location for the robotic arm base.

After extensive testing, the best stopping location is indicated in orange. When the robotic arm base is positioned at (15, 13), the coordinates of the cargo relative to the robotic arm become (1000, 1400, 200). In this configuration, there are no obstacles in the way, allowing the robotic arm to directly grab the goods. Therefore, (15, 13) is identified as the optimal stopping position for the robotic arm base.

3.3.4 Calculation of joint angles

Step one: Discretize the time interval into N time points, $t_k = k \cdot \Delta t$where k=0, 1, ..., n.

Step two: At each moment, calculate the corresponding joint angle through inverse kinematics$\theta_i(t_k)$,In order to maintain smooth motion, the interpolation algorithm can be used to smooth the changes in joint angles.

Step three: Use particle swarm algorithm to find the optimal joint angle.

Through the above algorithm and the multi-objective optimization model established in work 2, the minimum error at the end is 4.028×10-7mm, the minimum energy consumption is 23.9261 J, and the optimal joint angle is shown in Table 5:

Table 5 Optimized joint angles

Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
159.1864	-137.8819	35.44808	-116.0321	-82.45469	-112.2253

By analyzing Table 5 and the final results, it is evident that the joint rotation angles satisfy the rotation angle requirements for each joint as specified in the work. The minimum terminal error is extremely small, effectively meeting the robotic arm's requirements for picking up goods.

3.3 Model establishing and analyzing for work four

3.4.1 Model establishment

In this work, the task is to complete the grabbing of multiple targets while avoiding obstacles. To address this challenge, the Traveling Salesman work (TSP) can be applied [14]. Since TSP is an NP-hard work, a genetic algorithm is used to solve it. The genetic algorithm is employed to effectively find an approximately optimal path. Figure 9 presents a flowchart of the genetic algorithm used to solve the TSP work.

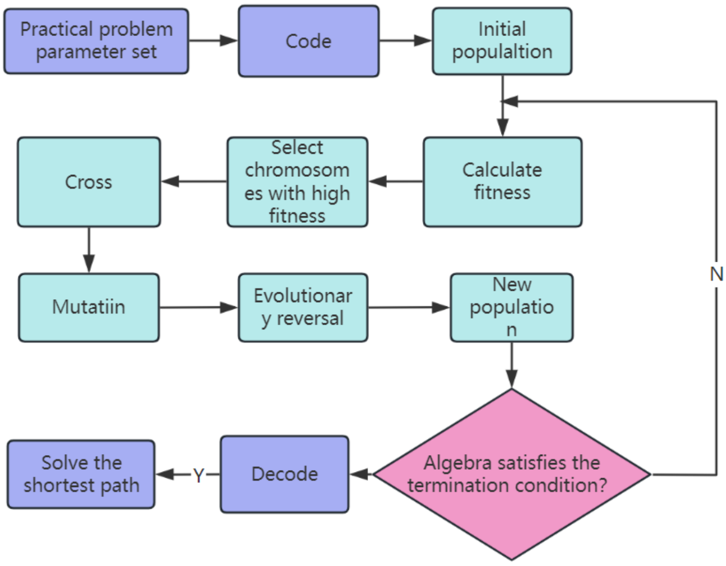

Figure 9 Genetic algorithm TSP work flow chart

Set the distance matrix D:

$$D_{ij} = distance(freight_i, freight_j) \tag{14}$$

Objective function:

$$minimize \sum_{i=1}^{n-1} D_i, i + 1D_{i+1} \tag{15}$$

The initial raster map is created by reading the content in the attachment, as shown in Figure 10:

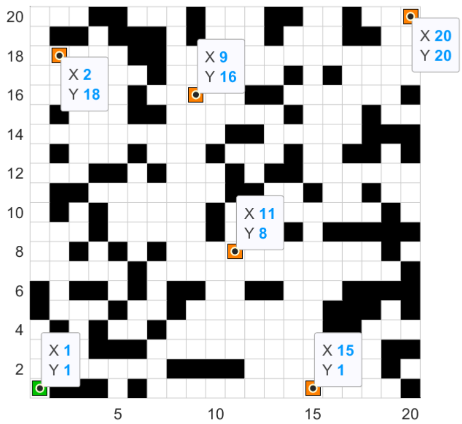

Figure 10 Original raster map

The grid map clearly shows the starting point of the robotic arm and the locations of each cargo. The green mark indicates the starting point, while the orange marks represent the five cargo target points. This setup simplifies the creation of the distance matrix for the TSP algorithm.

The multi-objective optimization model developed based on works two and three is as follows:

$$\min\left(\sqrt{(x-x_t)^2 + (y-y_t)^2 + (z-z_t)^2}, \sum_{i=1}^{6}\left(\frac{1}{2} I_i w_i^2\right)\right) \tag{16}$$

$$s.t\begin{cases}-160 \le \theta_1 \le 160\\ -150 \le \theta_2 \le 15\\ -200 \le \theta_3 \le 80\\ -180 \le \theta_4 \le 180\\ -120 \le \theta_5 \le 120\\ -180 \le \theta_6 \le 180\end{cases} \tag{17}$$

To address the multi-target picking task and ensure that the manipulator returns to its initial state after each operation, this article excludes the energy consumption associated with joint angle adjustments and focuses solely on the energy required for movement and picking. Subsequently, five joint angle optimizations are performed on the manipulator to minimize end error and energy consumption.

3.4.2 Model solution

The model is solved using the A* algorithm, with the results presented in Figures 11 and 12:

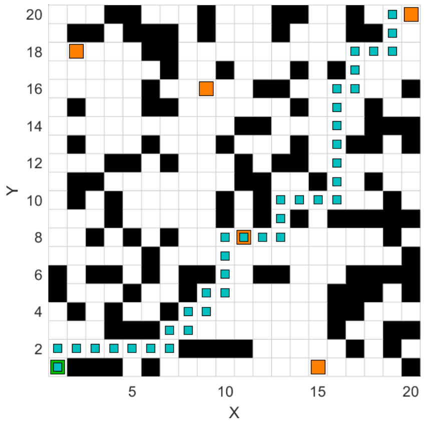

Figure 11 Return path planning

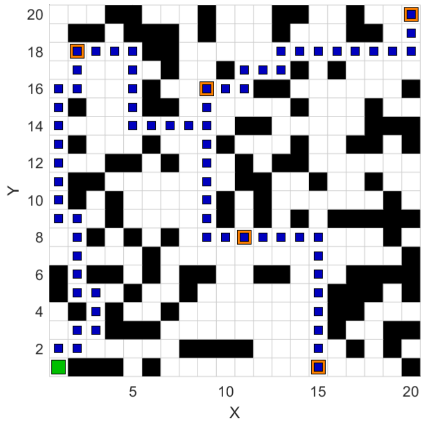

Figure 12 Pickup path planning

1.Figure 11 (Optimal Path Back to the Starting Point) illustrates the best route for the robotic arm to return to the starting point after completing the pickup task. By optimizing the movement path, the return path with the lowest energy consumption was identified.

2.Figure 12 (Shortest Optimized Path for Picking Up Goods) displays the shortest route for the robotic arm to pick up goods from the starting point to the target location. Through re-planning the target positions and optimizing the movement path, the picking path with minimal energy consumption was determined.

Based on the model from work three, considering the robotic arm's length, it is necessary to adjust the optimal stopping point to re-plan the target positions. Resetting the optimal stopping point ensures that the robotic arm performs the task with minimal energy consumption during both pickup and return processes. This optimization enhances the robotic arm's efficiency and extends its service life. Figure 13 illustrates the final result of this optimization.

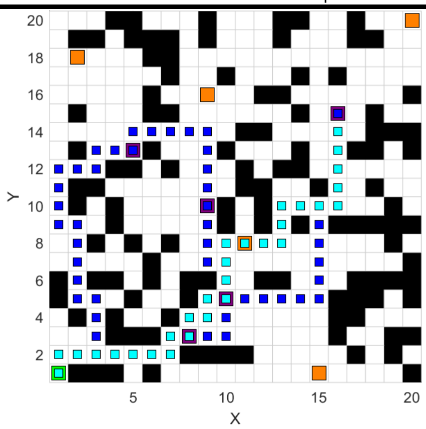

Figure 13 Complete path planning

In the picture, the red square represents the best stopping place of the robot arm, the orange square represents the target location, the blue square represents the pickup path, and the light blue square represents the best return path.

By analyzing Figure 13, the coordinates of the five optimal stopping points from the five cargo positions can be obtained. Subsequently, the terminal error, energy consumption and related optimal joint angles were solved through the genetic algorithm, and the results are shown in Table 6.

Table 6 Final solution results

Pick-up sequence	Cargo coordinate	Optimum stopping point	End error/mm	energy consumption/J	Optimal joint optimization Angle/°
1	(2，18)	(5,13)	197.4966	21.4476	64.1 -61.00 -188.9 -138.0 53.6 64.7
2	(9，16)	(9，10)	139.0754	21.8425	62.8 -88.5 -193.2 -150.9 -31.6 -32.6
3	(15，1)	(10，5)	161.2161	22.904	-10.6 -45.9 -195.3 -152.8 -58.6 2.37
4	(20，20)	(16，15)	70.625	22.1326	12.4 -90.1 -191.8 -145.1 -17.03 18.9
5	(11，8)	(8，3)	46.1155	23.5956	38.1 -10.1 -195.6 -150.7 -51.3 -38.3

Table 6 shows the detailed data of the optimal stopping point coordinates, terminal error, energy consumption and optimal joint angles for five cargo positions.

Through the analysis of the above table, it can be seen that the maximum terminal error is 197.4966mm, which is less than 200mm and meets the requirements of the work. The final energy consumption is 111.9223J, which is small. The optimization angle of each joint meets the multi-objective optimization conditions.

4 CONCLUSION

This model presents an effective solution for optimizing the design of manipulator joint angle paths. While it incorporates several simplifications and assumptions, its practical applications and potential for improvement remain significant. Future enhancements to the model's accuracy and practicality could be achieved by integrating more complex energy consumption models, addressing additional practical factors, and optimizing algorithm efficiency. The model offers robust methodological and theoretical support for optimizing robot arm motion in complex environments and is anticipated to find widespread use in fields such as industrial automation, precision manufacturing, and robotics.

5 ACKNOWLEDGEMENTS

This work was supported by national college student innovation and entrepreneurship training program project of Changzhou Institute of Technology(Grant No.: 202411055026Z) and ministry of education industry-university cooperative education project (Grant No.: 231106441092432)

REFERENCES

[1] Ye, T. (2023). Dynamic parameter identification and trajectory planning of six-degree-of-freedom humanoid manipulator. Nanchang University.

[2] Shi, L. C. (2015). Technical application and development of robots in the automation of plastic manufacturing. Electronic Test, (5), 131-134.

[3] Weixiao Q. Research on trajectory planning and target positioning methods of six-axis robotic arms. Hunan University, 2023.

[4] Panming, Z.，Wang, S.&Li, J. et al. (2024). Structural design and kinematic analysis of a 6-degree-of-freedom robotic arm. Mechanical Transmission, 48(06), 50-57.

[5] Chenyong G.Research on trajectory planning and optimization of six-degree-of-freedom manipulator. Changchun University of Technology, 2022.

[6] J. Denavit & R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," ASME Journal of Applied Mechanics, vol. 22, no. 4, pp. 215-221, 1955.

[7] Wang, Q. F. (2023). Dynamic modeling and high-precision control strategy of motion control system for six-axis robots in industrial production. China Machinery, 30, 35-38.

[8] Holland, J. H. (1975). *Adaptation in natural and artificial systems. University of Michigan Press.

[9] Guomeng S., Cao, Y.& Zhangya, B., et al. (2018). Research on path planning of planar robot manipulator based on interpolation algorithm. Electrical Technology, (18), 160-162.

[10] Randong, Ke., Pengfu, L.&Lihong, G. Review of path planning research based on A* algorithm. Electronic Technology and Software Engineering, 2020, 11(24): 11-12.

[11] Weifang, X., Sunming, G., & Zhouhai, Y. (2021). Kinematics analysis and verification of six-axis robotic arm based on MATLAB. Journal of Jilin Institute of Chemical Technology, 38(11), 59-62+79.

[12] Chenda, G., Liuy, X & Liuxing, D., et al. (2023). Motion analysis and trajectory planning of six-degree-of-freedom manipulator. Journal of Jilin Institute of Chemical Technology, 40(01), 80-86.

[13] Zhang, N. Intelligent optimization algorithm for solving nonlinear equations. Jilin University, 2013.

[14] Zhangsi, L. (2019). Research on the traveling salesman work with priority constraints. Logistics Engineering and Management, 41(12), 86-88+121.

[15] Rai, R., & Kumar, A. (2017). Robotics in manufacturing: A comprehensive review of applications and research. Manufacturing Technology, 117(4), 197-209.