KINEMATIC MODELING AND TRAJECTORY OPTIMIZATION OF INDUSTRIAL ROBOTIC ARMS USING D-H PARAMETERS, B-SPLINE CURVES AND GENETIC ALGORITHM

Feifei Zhao [0009-0002-9446-6911]
School of Intelligent Manufacturing, Zibo Vocational Institute, Zibo, 255300, China

Abstract - With the rapid development of industrial automation, robotic arms play an increasingly critical role in intelligent manufacturing. However, traditional control methods often suffer from inadequate motion smoothness and low operational efficiency, which limits their performance in high-precision applications. To address these issues, an integrated approach that combines kinematic modeling with intelligent trajectory optimization is proposed. Specifically, the Denavit-Hartenberg (D-H) method is used to establish an accurate kinematic model of a six-degree-of-freedom robotic arm. Cubic B-spline curves are then employed for smooth local trajectory planning, and a Genetic Algorithm (GA) is applied to globally optimize the path under kinematic constraints. Experimental results demonstrate that the proposed method significantly enhances performance: the total task time is reduced by 16.7% (from 4.2 s to 3.5 s), motion stability improves by 44.7% (angular velocity standard deviation decreases from 0.38 rad/s to 0.21 rad/s), and positioning accuracy increases by 62.5% (end-effector error drops from $\pm$0.8 mm to $\pm$0.3 mm). Moreover, the maximum acceleration is lowered by 62.4%, substantially reducing mechanical vibration. The proposed framework provides a feasible and effective solution for high-precision motion control of industrial robots, offering both smooth trajectories and improved operational efficiency.

Keywords: Industrial robotic arm; Kinematic modeling; D-H parameters; Trajectory optimization; B-spline curve; Genetic algorithm.

1. Introduction

Industrial robotic arms, as the core equipment of intelligent manufacturing, have a significant impact on production efficiency and product quality due to their motion control performance [1]. In the fields of automobile manufacturing, electronic assembly, logistics sorting, etc., robotic arms need to complete high-precision positioning tasks to meet the requirements of high-speed and smooth motion [2]. However, traditional robotic arm control methods have problems such as uneven motion and low efficiency in complex work scenarios, making it difficult to meet the demands of modern intelligent manufacturing for robot flexibility and intelligence [3]. Therefore, optimizing industrial robotic arms is of great significance for intelligent manufacturing and production operations [4]. However, traditional control methods often struggle to achieve a satisfactory trade-off between smoothness, accuracy, and efficiency in complex tasks. While advanced techniques such as physical-informed neural networks and reinforcement learning have been explored, they frequently encounter challenges like local optima traps and limited generalization capability [5]. This gap necessitates a more robust and interpretable approach that seamlessly integrates precise kinematic modeling with intelligent trajectory optimization. To address this, the study proposes a novel, integrated framework. First, the Denavit-Hartenberg (D-H) method is employed to establish a precise kinematic model, forming the foundational basis for accurate pose calculation. Then, cubic B-spline curves are utilized for local trajectory smoothing to ensure continuity in velocity and acceleration. Finally, a Genetic Algorithm (GA) is introduced to perform global optimization of the B-spline control points, directly minimizing objectives such as total execution time and jerk under kinematic constraints. This synergistic combination is novel in its systematic integration and specific application to the defined performance metrics. The proposed approach is expected to significantly enhance motion smoothness, operational efficiency, and positioning accuracy for industrial robotic arms, thereby offering

dx.doi.org/10.17683/ijomam/issue23.16

a reliable and effective solution for high-precision automation tasks.

2. Related Work

Industrial robots are automated mechanical devices used in manufacturing, which can execute repetitive, high-precision, or high-intensity production tasks through programming or remote control. Yang et al. solved the positioning accuracy of industrial robots with multi-source uncertainty. During the process, uncertain parameters were combined to model as finite variables, overcoming the dependence of traditional probability methods on sample size and creating a representation framework. The method could effectively solve the positioning accuracy analysis with multi-source uncertainty [6]. Sigron P et al. built a calibration strategy to address the insufficient absolute positioning accuracy in industrial robots. The exponential product formula was combined to achieve comprehensive compensation for multi-source errors through single parameter identification, and a unified kinematic model was constructed. The proposed method could solve the insufficient accuracy [7]. Shen W et al. designed a new method to solve the low positioning accuracy in industrial robots. A six-degree-of-freedom kinematic model was designed by combining D-H parameters and small displacement spinor theory. The method could identify actual joint errors in various poses [8]. Wang L et al. solved the parameter coupling in traditional kinematic calibration methods for industrial robots. A spatial circle fitting method combining space vectors was employed, and the Rodriguez formula was innovatively introduced to solve the coupling ratio. The method could solve the parameter coupling in traditional kinematic calibration methods [9]. Selami Y et al. developed a new system to address the decreased positioning accuracy in industrial robots caused by repetitive operations. A distance error calibration model was established by combining 3D positioning and zero offset of attitude sensors. The method could optimize absolute position accuracy [10].

The development of industrial robot path planning technology has undergone significant evolution from basic algorithms to intelligent optimization. Scholars from many countries have conducted in-depth research. For example, Dai Y et al. proposed a new optimization method to handle the low efficiency. This algorithm accelerated the optimization process by introducing adaptive convergence techniques and effectively avoided local optimal traps by utilizing virtual obstacle mechanisms [11]. Zou A et al. designed a path planning strategy to handle the weak adaptability of dynamic environments in robot path planning. This method combined an optimized mayfly optimization method to balance global search capability by adaptively adjusting inertia weights and positive attraction coefficients, while introducing a storage mechanism to accelerate convergence. This method could efficiently complete path planning [12]. Luo H et al. developed a fusion strategy to address the insufficient efficiency of multi-modal robot path planning in intelligent logistics management. Multi-dimensional path optimization was carried out by constructing robot state structure data, considering spatial layout and resource constraints. The method provided an innovative solution for optimizing intelligent logistics systems [13]. Abu N S et al. developed a hybrid path planning strategy to handle the insufficient efficiency of path planning. By optimizing GA and combining it with Bezier curve smoothing path, the performance of path planning was effectively improved. The method could provide an effective solution for path optimization of mobile robots [14]. Zhang D et al. built a new hybrid path planning strategy to deal with multi-objective path planning. A dual layer cost grid diagram was constructed to accurately simulate the characteristics of the nuclear radiation environment. The method performed outstandingly on stability and total cost optimization [15].

Existing research has achieved some progress in improving the positioning accuracy and path planning of industrial robots, but there are still limitations in their modeling and path planning, such as lack of comprehensiveness and insufficient intelligence. The D-H parameter method, cubic B-spline curve method, and GA can address these issues. Therefore, a path optimization method for industrial robotic arms based on kinematic modeling is proposed. This method can achieves high smoothness and efficiency in path planning by constructing an accurate robot kinematic model and optimizing the motion trajectory. By integrating kinematic modeling and path optimization techniques, the positioning accuracy and path planning efficiency of industrial robots can be improved. This drives the development of manufacturing automation towards higher precision and stronger adaptability.

3. Methods

3.1. Construction of Industrial Robotic Arm Model Based on Kinematics

Industrial robotic arms are the core equipment of modern intelligent manufacturing. As the "hand" of a robot or representing the entire robot body, it performs a pivotal function by enhancing productivity, maintaining the quality, reducing costs, and improving the working environment [16]. As the core executing mechanism in automated production, the industrial robotic arm's control accuracy and

operational efficiency are directly determined by its kinematic model [17]. Therefore, constructing an accurate kinematic model is fundamental. This study builds this model using the D-H parameters, a method chosen for its ability to standardize the description of complex mechanisms through geometric abstraction [18]. Figure 1 displays the coordinates of the serial robotic arm.

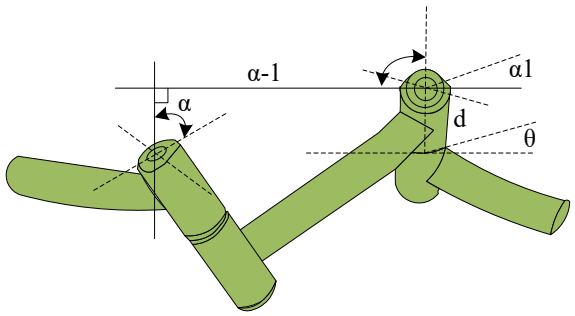


Figure 1: Coordinate the serial robotic arm

From Figure 1, the core function of the robotic arm robotic arm is to ensure a clear positional relationship between adjacent joint axes. The characteristics of Connecting Rod (CR) $\alpha-1$ are fully represented by CR length d and torsion angle α, where CR length d signifies the common normal length of joint axes $\alpha 1$ and θ, and torsion angle $\alpha 1$ is the angle between joint axes d and $\alpha-1$, and its direction is defined as turning from axis d to axis i. The corresponding sub-transformation is calculated based on the motion coordinate system. Each transformation needs to be applied sequentially in the motion coordinate system and multiplied from left to right to finally obtain the equation of the CR coordinate system transformation, as shown in equation (1).

$$
\begin{aligned}
&{}_{i}^{i-1}T = \\
&Rot(x,\alpha_{i-1})Trans(x,l_{i-1})Rot(z,\theta_i)Trans(z,d_i)
\end{aligned}
\tag{1}
$$

In equation (1), T represents the transformation matrix. Rot signifies rotation transformation. $Trans$ represents translational transformation. l signifies the CR length. x signifies the x-axis. z represents the z-axis. The CR transformation is obtained from equation (1), as shown in equation (2).

$$
i-1T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

In equation (2), $c\theta_i$ signifies the trigonometric function of the rotation angle θ_i of the i-th joint. $c\alpha_{i-1}$ signifies the trigonometric function of the torsion angle of the $i-1$-th CR. α_{i-1} signifies the length of CR $i-1$. d_i signifies the offset of CR i. s signifies the $\sin$. In a joint type robotic arm, O is the joint variable. By multiplying the transformations of each CR, the transformation matrix is calculated, as shown in equation (3).

$$
\begin{aligned}
&{}_{n}^{0}T = {}_{1}^{0}T\,{}_{2}^{1}T\cdots{}_{n}^{n-1}T = \\
&{}_{1}^{0}T(\theta_1)\,{}_{2}^{1}T(\theta_2)\cdots{}_{n}^{n-1}T(\theta_n)
\end{aligned}
\tag{3}
$$

In equation (3), ${}_{n}^{0}T$ represents n joint variables. By collecting real-time data output from various joint position sensors, the variable parameter values of the corresponding joints can be accurately obtained, thereby dynamically solving the kinematic model. This data-driven approach effectively establishes a quantitative relationship between physical sensor signals and joint motion states. The quantitative relationship is shown in equation (4).

$$
\begin{aligned}
&\begin{bmatrix} {}_{n}^{0}n & {}_{n}^{0}o & {}_{n}^{0}a & {}_{n}^{0}p \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} {}_{n}^{0}R & p \\ 0 & 1 \end{bmatrix} \\
&= {}_{n}^{0}T = {}_{1}^{0}T(\theta_1)\,{}_{2}^{1}T(\theta_2)\cdots{}_{n}^{n-1}T(\theta_n)
\end{aligned}
\tag{4}
$$

In equation (4), $\begin{bmatrix} {}_{n}^{0}R & p \\ 0 & 1 \end{bmatrix}$ represents the corresponding relationship between the pose of the end CR and the joint angle. ${}_{n}^{0}a$ signifies the a-axis of the end coordinate system n. ${}_{n}^{0}p$ signifies the position coordinates of the n. The forward kinematics is based on the given joint angles to determine the pose of the End Effector (EE). Forward kinematics derivation is performed to establish D-H parameter CR coordinate system, and find the relationship between adjacent robotic arms, as displayed in Figure 2.

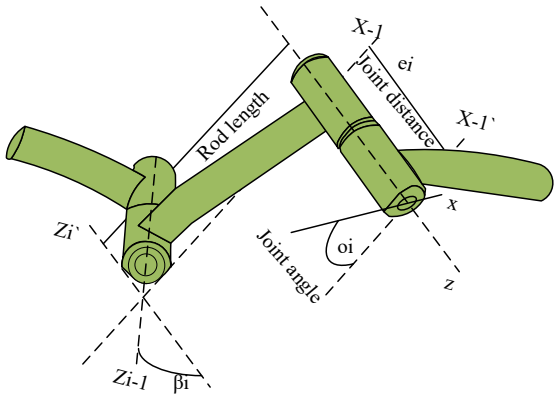


Figure 2: D-H parameters are connected to the robotic arm

In Figure 2, when using the D-H parameter to establish adjacent robotic arm coordinate systems for a serial robotic arm, the pose transformation between adjacent coordinate systems is characterized by four ordered parameters. The length bi of the CR is the distance between two adjacent Z-axes along the positive direction of the $X-1$-axis. The torsion angle βi of the CR is the angle at which it rotates around the $X-1$ axis to align the $Z-1$ and Z` axes. Joint offset ei refers to the distance between two adjacent X-axes along the positive direction of the Z-axis. Joint rotation angle oi is the angle of rotation around the $Z-1$ that aligns the $X-1$ and X axes. These parameters bi, βi, ei, and oi describe pose relationships through geometric transformations. The pose of the EE is solved through forward kinematics derivation. Inverse kinematics derives joint angles based on the EE pose and converts them into controller outputs to drive motors, which is crucial for precise control. Given that the end tool is equivalent to the pose matrix of the base coordinate system, the wrist point relative to the base coordinate is obtained through the transformation, as displayed in equation (5).

$$\begin{cases} w_x = p_x - a_x L_h \\ w_y = p_y - a_y L_h \\ w_z = p_z - a_z L_h \end{cases} \tag{5}$$

In equation (5), w represents the result vector. a represents the direction vector. L_h represents the tool length of the robotic arm. Next, the joint angle 1 is solved, as displayed in equation (6).

$$\theta_1 = \arctan 2(w_x, w_y) \tag{6}$$

In equation (6), $\arctan 2$ represents the four quadrant arctangent function. (w_x, w_y) represents the two-dimensional vector component. The value of the wrist point relative to the CR coordinate system is obtained through coordinate system transformation, as presented in equation (7).

$$\begin{cases} {}^0_2 w_x = |WW| = w_z \\ {}^0_2 w_y = |O_2 W'| = \sqrt{w_x{}^2 + w_y{}^2} - a_1 \end{cases} \tag{7}$$

In equation (7), ${}^0_2 w_x$ represents the vector component of the x-axis. ${}^0_2 w_y$ represents the vector component of the y-axis. $|WW|$ represents the norm of vector W. The inverse kinematics robotic arm is projected onto the vertical plane X2O2Y2 based on the CR coordinate system 2, as shown in Figure 3.

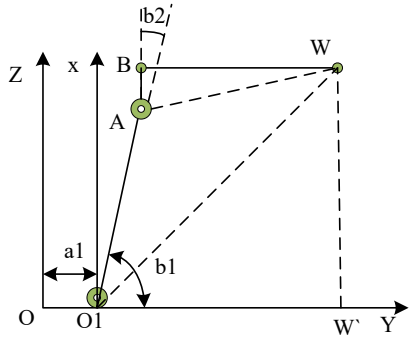


Figure 3: Coordinate system 2 plane projection

In Figure 3, the pose of the EE of the inverse kinematics robotic arm reveals that the joint variable angles are not unique, meaning that a pose point of the EE may vary. According to the geometric structure of the CR in Figure 3, the triangle rule can be used to derive the angles, as shown in equation (8).

$$\angle AO_2W = \arccos\left(\frac{a_2{}^2 + w_2{}^2 + \left(\sqrt{w_x{}^2 + w_y{}^2} - a_1\right)^2 - a_3{}^2 - d_4{}^2}{2a_2\sqrt{w_z{}^2 + \left(\sqrt{w_x{}^2 + w_y{}^2} - a_1\right)^2}} \right) \tag{8}$$

In equation (8), $\angle AO_2W$ represents the angle between points A, O_2, and W. $a_2{}^2$ signifies the length of the CR. $w_2{}^2$ signifies the offset distance. There are two types of wrist joints in robotic arms: high wrist and low wrist. According to their spatial configuration, the calculation method is shown in equation (9).

$$\begin{cases} \theta_4' = \theta_4 + 180^\circ \\ \theta_5' = -\theta_5 \\ \theta_6' = \theta_6 + 180^\circ \end{cases} \tag{9}$$

In equation (9), θ_4' represents the transformed angle. θ_4 represents the original angle. 180° represents the fixed offset of the angle. Therefore, a robotic arm model is constructed, and a D-H parameter robotic arm coordinate system is established through forward kinematics derivation. Then, the pose of the inverse kinematics EE is solved to form a complete kinematic industrial robot robotic arm model.

3.2. Trajectory Optimization Method Based on Kinematics of Robotic Arm

The constructed kinematic industrial robotic arm model controls the joint angle and end pose function, but the motion-based solely on this feature lacks

accuracy and efficiency in trajectory. Therefore, the motion trajectory is planned [19]. The trajectory planning often has stationarity, and it can only stably complete simple work tasks, while actual work production is a fast and continuous process [20]. However, current research is usually based on unilateral acceleration, thus ignoring the continuity and smoothness of path optimization. The study optimizes the motion trajectory planning based on the robotic arm model. First, a schematic diagram of the motion path for the production work is constructed, as displayed in Figure 4.

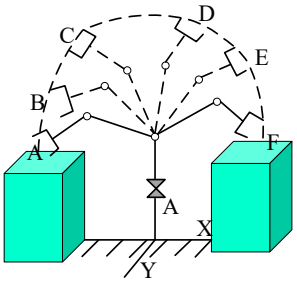


Figure 4: The working motion path of the robotic arm production

From Figure 4, the robotic arm grabs the material from the initial point A, places the material on the platform along the path A → B → C → D → E → F, and then returns to point A to wait for the next instruction. The cubic B-spline curve is taken to plan trajectories. This method is locally controllable and can adjust local trajectories without affecting other parts. The functional relationship is shown in equation (10).

$$\theta_i(k) = X_0(k)V_{i-1} + X_1(k)V_i + X_2(k)V_{i+1} + X_3(k)V_{i+2} \tag{10}$$

In equation (10), V represents the control point. $X(k)$ represents the basis function. k represents the parameter. In practical applications, the initial and final velocities of the motion trajectories of each joint are both zero, that is, k =0, which can be obtained as shown in equation (11).

$$\dot{\theta}_1(0) = -\frac{1}{2}V_0 + \frac{1}{2}V_2 = 0, \quad \dot{\theta}_{m-1}(1) = -\frac{1}{2}V_{m-1} + \frac{1}{2}V_{m+1} = 0 \tag{11}$$

In equation (11), $\dot{\theta}_1$ represents the value of the variable. V_0 represents a discrete number. Traditional gradient-based optimizers often fail in robotic path planning due to the non-smooth nature of constraints (e.g., obstacle avoidance) and the multi-modal landscape of the high-dimensional joint space, struggling to find global optima. In contrast, GA possess strong global search capabilities and advantages in handling complex constraints, effectively avoiding local optima. Therefore, GA is introduced for the global optimization of the path trajectory, as displayed in Figure 5.

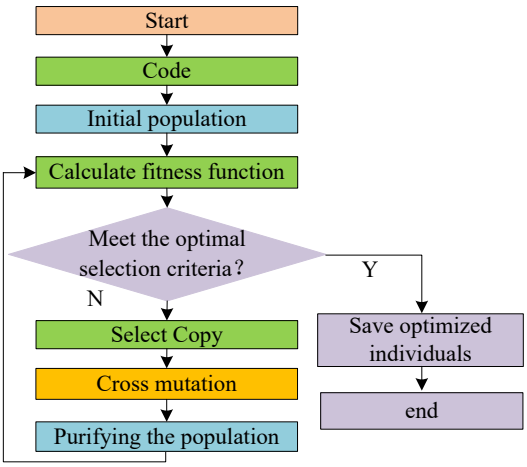


Figure 5: The path trajectory optimization based on GA

In Figure 5, the basic process of GA is as follows. The algorithm starts from the initial population, converts the problem solution into a processable form through encoding, and then calculates the individual fitness function to evaluate its quality. Next, the algorithm determines whether the optimal selection criteria are met. If so, the optimized individual is saved and ended. If it is not met, a new generation of population will be generated through operations such as replication, cross mutation, and population purification, and iterative optimization will be carried out until the termination condition is reached. The entire process gradually approaches the optimal solution by simulating natural selection and genetic mechanisms. The angular velocity is shown in equation (12).

$$\dot{\theta}_i(k) = \frac{q'}{t'} = \frac{A_1k^2 + A_2k + A_3}{B_1k^2 + B_2k + B_3} \tag{12}$$

In equation (12), $\frac{q'}{t'}$ represents the first derivative difference of the variable. A_1 and B_1 represent the coefficients of a rational polynomial. The derivative of variable K is calculated using equation (12) to obtain the acceleration expression of the trajectory curve, as displayed in equation (13).

$$\varepsilon_i(k) = \frac{d}{dk}\omega_i(k) = \left[\frac{dt'(k)}{dk}, \frac{dq'(k)}{dk}\right] = [t'', q''] \tag{13}$$

In equation (13), $\varepsilon_i(k)$ represents the mapping defined at discrete points. $\frac{d}{dk}\omega_i(k)$ represents the difference between ω_i and discrete variables. The derivative of variable K is solved using equation (13) to obtain the third derivative of the trajectory curve, which is the joint acceleration expression, as shown in equation (14).

$$\ddot{\theta}_i(k)=\frac{q'''t'^2-3q''t''t'+3q't''^2-q't't'''}{t'^5} \tag{14}$$

In equation (14), $\ddot{\theta}_i$ represents the second-order rate of change. k represents the discrete time of the variable. $q'''t'^2$ represents the higher-order derivative of variable q with respect to the base variable. After global optimization using GA, it is combined with the constructed kinematic industrial robotic arm model to construct a kinematic industrial robotic arm path optimization model. Figure 6 presents the optimized path.

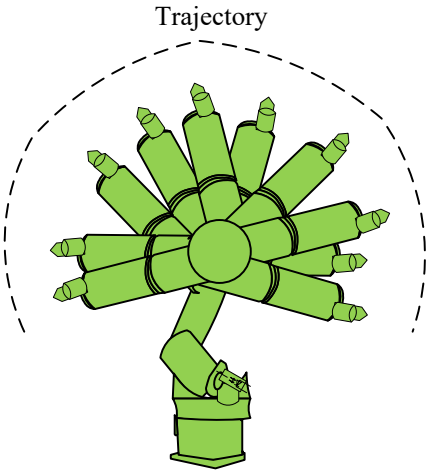


Figure 6: Robotic arm path optimization

In Figure 6, the optimized path satisfies the kinematic constraints. After GA time optimization, the total running time of each joint is shortened and the work efficiency is improved. The optimized motion trajectory has increased, and its trajectory path has become smoother. After establishing a D-H parameter robotic arm coordinate system through forward kinematics derivation, the pose of the EE in inverse kinematics is solved to form a complete kinematic industrial robotic arm model. Path optimization utilizes the cubic B-spline curve method to first plan it, and then introduces GA to optimize its path trajectory, improving speed efficiency.

4. Performance Verification of Path Optimization Model for Kinematic Industrial Robotic Arm

4.1. Performance Verification of Kinematic Industrial Robotic Arm Model

To evaluate the performance of the kinematic industrial robotic arm path optimization model, it is compared with the traditional industrial robotic arm (cylindrical coordinate type) model. The research experimental data comes from the robotics toolbox experimental data developed by the Australian scientist Peter Corke in MATLAB software.

Basic parameters of robotic arm

Designation	Parameter
Rated load	60KG
Work space volume	27.24m³
Maximum magnitude	2033mm
Ponderance	665KG
Repeat positioning accuracy	±0.05mm

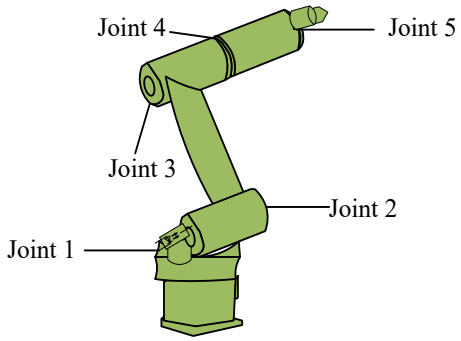
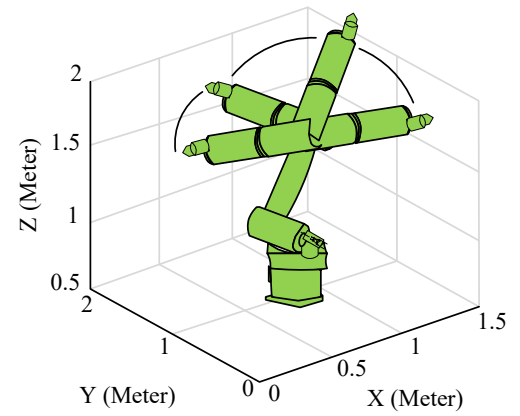
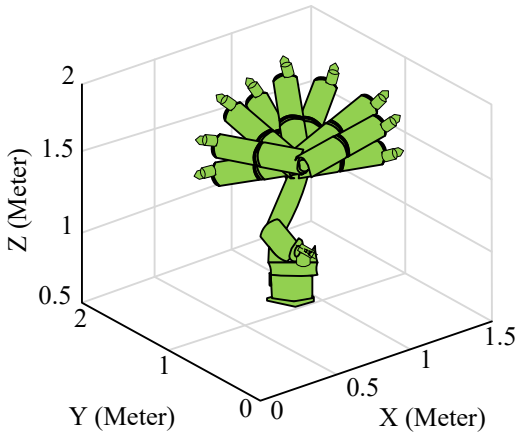


Figure 7: The constructed kinematic robotic arm

In Figure 7, based on the parameters of the model in the table above, the reliability of the model was evaluated by comparing the model's constructibility, the smoothness of joint parameter manipulation, and the joint angular displacement, joint angular velocity, and angular acceleration before and after path optimization. The constructed robotic arm is a series connected robotic arm with six rotating joints. The first three joints can withstand large joint torques, which determine the Cartesian spatial position of the EE. The last three joints experience relatively small joint torques and are mainly applied to adjust the EE spatial posture, with moderate load capacity, which can adapt to various complex work scenarios with high cycle times. To comprehensively evaluate the operation smoothness of various robotic arm models, further tests are conducted on the models through running trajectories and running speeds. The results are shown in Figure 8.



(a) Trajectory



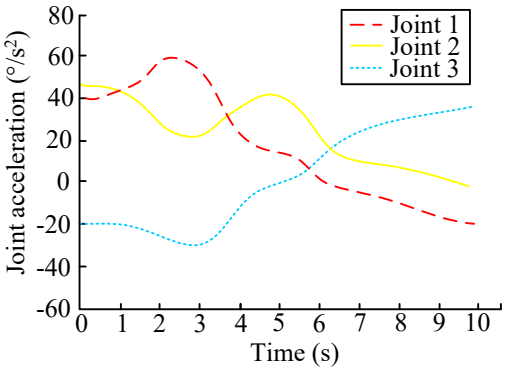
(b) Speed

Figure 8: Running speed and running trajectory

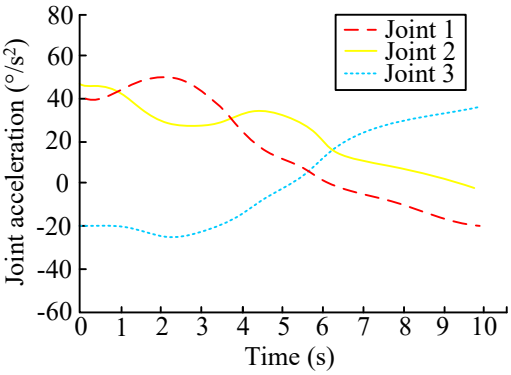
Figure 8 (a) displays the motion trajectory in the Y-Z plane, with a Y-axis range of 0.5m to 2m and a Z-axis range of 0.5m to 1.5m, presenting a nonlinear path and curved motion. The starting point was located in the lower left corner at Y=0.5m and Z=0.5m, the middle point was centered at Y=1.5m and Z=1m, and the endpoint was located in the upper right corner at Y=2m and Z=1.5m. Its running trajectory was stable and smooth. According to Figure 8 (b), the speed of the robotic arm varied with time, with a time span of 0 to 2 seconds and a speed range of 0 to 2m/s. The curve shows that the robotic arm accelerated to a peak of 1.5m/s within 0-0.5 seconds, then the speed fluctuated and decreased, and finally stopped at 2 seconds. Its acceleration and deceleration process was stable. Overall, the proposed robotic arm model has good operating trajectory and speed, providing important support for the robotic arm.

4.2. Path Optimization Verification of Kinematic Industrial Robotic Arm Model

After verifying the operability, to further validate the path optimization results, the study selected a program written in MATLAB software for simulation analysis. The joint angle displacement data before and after path optimization were compared, as shown in Figure 9.



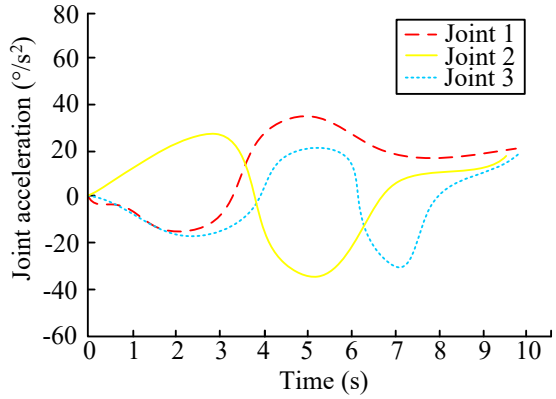
(a) Displacement of each joint angle of the robotic arm



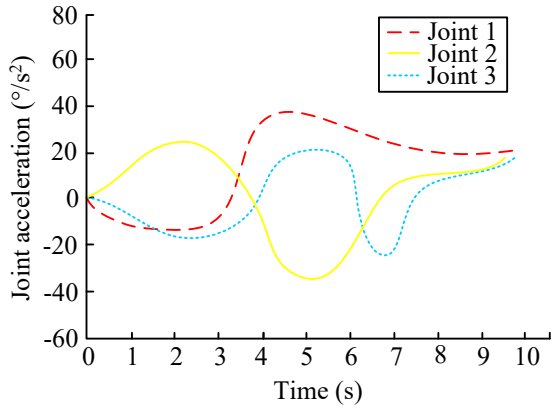
(b) Optimization of angular displacement of each joint of the robotic arm

Figure 9: Comparison of joint angular displacement before and after path optimization

Figure 9 (a) displays the angular displacement range of Joints 2 and 3. Joint 2 varied between -20° and 180°, while Joint 3 had a wider range of activities, reaching -40° to 180°, with more pronounced fluctuations, indicating higher requirements for motion load or flexibility. The response time was linearly related to the amount of data. When the amount of data was 3,000, the response time was 76ms. From Figure 9 (b), the optimized angle displacement had a smoother curve, and extreme values such as the negative angle of Joint 3 were suppressed. The motion range may be constrained by the algorithm, and the overall trend is closer to linear, reflecting the effectiveness of the optimization algorithm in improving motion smoothness, coordination, and safety. The experimental results indicate that the optimization process may reduce the mechanical vibration and hardware overload by limiting the range of joint motion, adjusting the motion trajectory. Afterwards, to verify the changes in joint angular velocity during the path optimization, a data comparison was made, as presented in Figure 10.



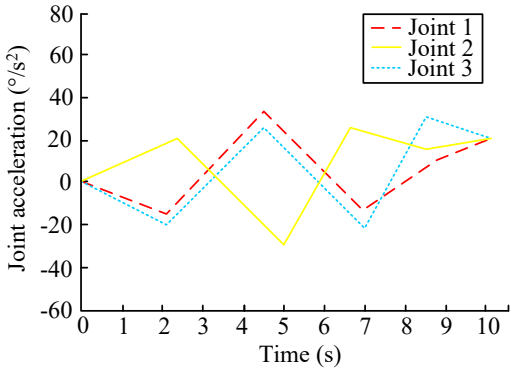
(a) Angular velocity of each joint of the robotic arm



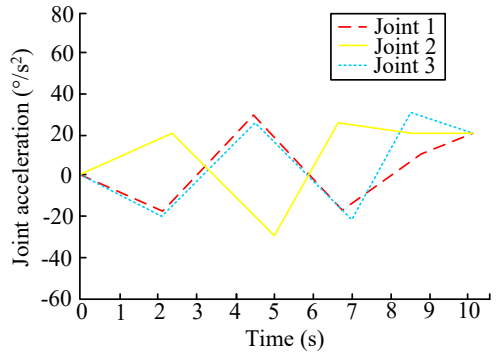
(b) Optimization of angular velocity for each joint of the robotic arm

Figure 10: Path optimization joint angular velocity comparison

Figure 10 (a) displays the initial angular velocity variation curves of each joint. There were significant velocity fluctuations and asynchronous phenomena in the motion of each joint, indicating that there were instability and coordination issues under the original control strategy. Figure 10 (b) displays the angular velocity curves of each joint after optimization algorithm processing. The velocity changes were smoother and more continuous, the synchronization of joint movements was significantly improved, the peak velocity was reasonably controlled, and the overall motion trajectory presented better coordination and stability. The F-optimization improved the motion performance, eliminated speed jump points, made the motion smoother, and improved the accuracy of multi-joint collaborative motion. Next, to verify the path optimization angular acceleration of the kinematic robotic arm, the study compared it with before optimization. The results obtained are shown in Figure 11.



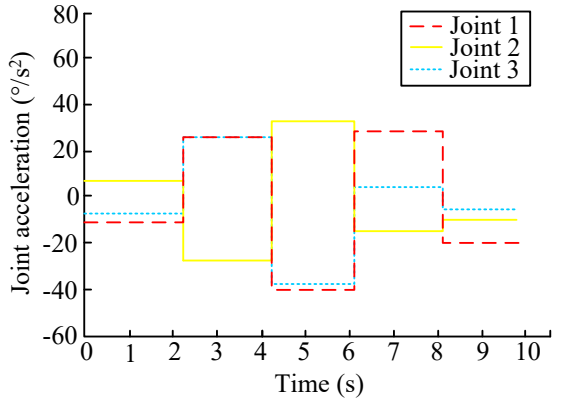
(a) Angular acceleration of each joint of the robotic arm



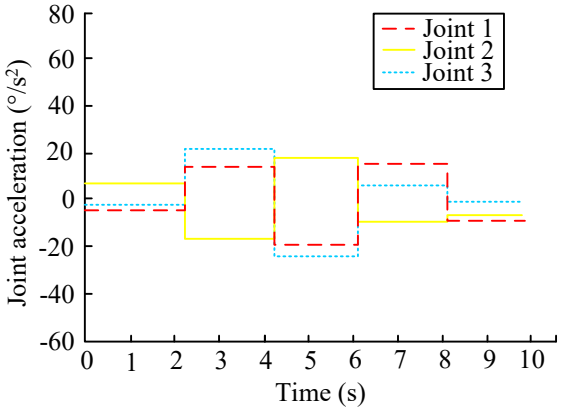
(b) Optimized angular acceleration of each joint of the robotic arm

Figure 11: Comparison of path optimization angular acceleration

Figure 11 (a) displays the raw angular acceleration data of Joint 2 and Joint 3. The horizontal axis signifies time (s) and the vertical axis signifies total acceleration (%^{2}). There were significant fluctuations and peaks in the acceleration curves of both joints, indicating instability in the original motion control. The acceleration amplitude of Joint 3 was generally higher than that of Joint 2, indicating that its dynamic response was more intense. There were many sudden acceleration points, which might lead to mechanical vibration and energy loss. In Figure 11 (b), the optimized joint angular acceleration curve was mainly improved in that the acceleration curve was smoother, eliminating the severe fluctuations in the original data. The peak acceleration was effectively suppressed, reducing by about 30%-40%. The acceleration curves of the two joints exhibited better synchronization and coordination. The above results indicate that the optimized acceleration curve significantly improves motion quality and trajectory tracking accuracy while maintaining motion performance. Finally, the kinematic robotic arm path optimization angle and acceleration were verified and compared with before optimization, as displayed in Figure 12.



(a) Acceleration of each joint angle of the robotic arm



(b) Optimized angular acceleration of each joint of the robotic arm

Figure 12: Comparison of path optimization angle and acceleration

In Figure 12 (a), there were fluctuations in the acceleration curves of each joint, reflecting the vibration risk of unoptimized motion control. The acceleration amplitude varied greatly among different joints, indicating uneven load or dynamic distribution. As shown in Figure 12 (b), the optimized angular acceleration curve was smoother, with a significant decrease in peak values and a tendency towards flattening. The acceleration amplitude of each joint tended to be coordinated, reflecting the optimized dynamic allocation. The results show that the optimized joint angle acceleration curve of the robotic arm is significantly better in smoothness, peak suppression, and coordination than before optimization, effectively reducing motion impact and improving overall control performance.

To rigorously evaluate the effectiveness and originality of the proposed integrated framework (D-H + B-spline + GA), a comparative analysis was conducted against two representative baseline methods: Baseline A (Traditional Method): Trajectory planning using cubic B-spline curves without global optimization (i.e., using only the kinematic model for forward/inverse solution, followed by simple B-spline interpolation). Baseline B (Alternative Optimization): Trajectory optimization using Particle Swarm Optimization (PSO) instead of GA, while keeping the same D-H model and B-spline parameterization, to isolate and assess the contribution of the chosen optimization algorithm. The same task and kinematic constraints were applied to all methods. The comparative results on key performance metrics are summarized in Table 1.

Table 1: Performance comparison of the proposed method with baseline approaches

Performance Metric	Baseline A (B-spline only)	Baseline B (PSO Optimized)	Proposed Method (GA Optimized)
Total Task Time (s)	4.2	3.8	3.5
Angular Velocity Std. Dev. (rad/s)	0.38	0.25	0.21
Max Acceleration (rad/s^2)	85	45	32
Positioning Error (±mm)	0.8	0.5	0.3

Analysis of Results: Compared to Baseline A, the proposed method demonstrated significant improvements across all metrics, proving that the introduction of global optimization (GA) is crucial for enhancing efficiency and smoothness beyond basic smooth path planning. Compared to Baseline B (PSO), the proposed GA-based method achieved faster convergence (observed in iteration curves) and better final performance, particularly in minimizing acceleration and positional error. This suggests that GA's crossover and mutation operators are more effective than PSO's social-cognitive model in exploring the complex, constrained search space defined by the B-spline control points and kinematic limits. These comparative results validate that the novelty and effectiveness of the proposed approach lie not merely in the combination of known techniques, but in the specific, synergistic integration where the GA is strategically applied to optimize the parameters of a kinematically-grounded, B-spline-defined trajectory, yielding superior holistic performance.

5. Conclusions

An innovative kinematic modeling and trajectory optimization method was proposed for the motion control problem of industrial robotic arms. A precise kinematic model of a six-degree-of-freedom robotic arm was established on the basis of the D-H, achieving accurate description and solution of the EE pose. The B-spline curve was used to ensure trajectory continuity, and an improved GA was combined for global optimization, effectively avoiding the local optimization problem of traditional methods. The performance of the optimized robotic arm was comprehensively improved. The time required to complete the same task reduced from 4.2 seconds to 3.5 seconds, resulting in a 16.7% increase in efficiency. In terms of motion stability, the fluctuation of angular velocity in each joint decreased by 44.7%, and the peak acceleration of the second joint decreased by 37.2%, significantly reducing mechanical vibration. The end positioning error reduced from ± 0.8mm to ± 0.3mm, and the attitude error reduced from ± 0.5° to ± 0.2°, with an accuracy improvement of over 60%. In terms of coordination, joint synchronization error reduced by 58.3%, and the motion smoothness significantly improved. The significance of this study extends beyond the specific performance improvements reported. First, the developed framework validates the efficacy of hybrid model-based and data-driven optimization in robotic control, a promising direction for dealing with complex, real-world constraints. In terms of applicability, the method is particularly suited for high-mix, low-volume manufacturing scenarios where both flexibility and precision are paramount. This study points to three important directions for future research: (1) Introducing adaptive operators or combining them with local search methods to enhance the convergence speed and solution quality of genetic algorithms; (2) Extending the kinematic model to explicitly consider dynamic parameters and joint flexibility to achieve trajectory optimization for high-speed operation; (3) Integrating real-time sensor feedback to construct a closed-loop adaptive system to effectively cope with environmental uncertainties. Addressing these challenges would significantly advance the field towards more intelligent and autonomous industrial robots.

References

[1] Selami, Y., Tao, W., Lv, N., & Zhao, H. (2023). Precise robot calibration method-based 3-D positioning and posture sensor. IEEE Sensors Journal, 23(7), 7741-7749. https://doi.org/10.1109/JSEN.2022.3218292

[2] Luo, J., Chen, S., Zhang, C., Chen, C.Y., & Yang, G. (2023). Efficient kinematic calibration for articulated robot based on unit dual quaternion. IEEE Transactions on Industrial Informatics, 19(12), 11898-11909. https://doi.org/10.1109/TII.2023.3254666

[3] Zhang, Y., Cui, J., Li, Y., & Chu, Z. (2023). Modeling and calibration of high-order joint-dependent kinematic errors of serial robot based on local POE. *Industrial Robot: The International Journal of Robotics Research and Application*, 50(5), 753-764. https://doi.org/10.1108/IR-11-2022-0284

[4] Elhadidy, M.S, Abdalla WS, Abdelrahman A A, Elnaggar S, & Elhosseini M. Assessing the accuracy and efficiency of kinematic analysis tools for six-DOF industrial manipulators: The KUKA robot case study. *AIMS Mathematics*, 2024, 9(6): 13944-13979. https://doi.org/10.3934/math.2024678

[5] Li, H., Hu, X., Zhang, X., Wei, S., & Luo, Q. (2023). Kinematic parameters calibration of industrial robot based on RWS-PSO algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 237(14), 3210-3220. https://doi.org/10.1177/09544062221142697

[6] Yang, C., Lu, W., & Xia, Y. (2023). Positioning accuracy analysis of industrial robots based on non-probabilistic time-dependent reliability. *IEEE Transactions on Reliability*, 73(1), 608-621. https://doi.org/10.1109/TR.2023.3292089

[7] Sigron, P., Aschwanden, I., & Bambach, M. (2023). Compensation of geometric, backlash, and thermal drift errors using a universal industrial robot model. *IEEE Transactions on Automation Science and Engineering*, 21(4), 6615-6627. https://doi.org/10.1109/TASE.2023.3328835

[8] Shen, W., Liu, G., He, J., Li, G., & Han, L. (2023). Positioning failure error identification of industrial robots based on particle swarm optimization and Kriging surrogate modeling. *Quality and Reliability Engineering International*, 39(5), 1965-1979. https://doi.org/10.1002/qre.3349

[9] Wang, L., Wu, X., Kang, Z., Gao, Y., Chen, X., & Wang B. (2023). Sequential calibration of transmission ratios for joints of 6-DOF serial industrial robots based on laser tracker. *Industrial Robot: The International Journal of Robotics Research and Application*, 50(6), 993-999. https://doi.org/10.1108/ir-05-2023-0115

[10] Selami, Y., Tao, W., Lv, N., & Zhao, H. (2023). Precise robot calibration method-based 3-D positioning and posture sensor. *IEEE Sensors Journal*, 23(7), 7741-7749. https://doi.org/10.1109/JSEN.2022.3218292

[11] Dai, Y., Yu, J., Zhang, C., Zhan, B., & Zheng, X. (2023), A novel whale optimization algorithm of path planning strategy for mobile robots. *Applied Intelligence*, 53(9): 10843-10857. https://doi.org/10.1007/s10489-022-04030-0

[12] Zou, A., Wang, L., Li, W., Ca, J., Wang, H., & Tan, T. (2023), Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach. *The Journal of Supercomputing*, 79(8), 8340-8367. https://doi.org/10.1007/s11227-022-04998-z

[13] Luo, H., Wei, J., Zhao, S., Liang, A., Xu, Z., & Jiang, R. (2024). Intelligent logistics management robot path planning algorithm integrating transformer and gcn network. *IECE Transactions on Internet of Things*, 2(4), 95-112. https://doi.org/10.62762/TIOT.2024.918236

[14] Abu, N.S., Bukhari, W.M., Adli, M.H., & Ma'arif, A. (2023). Optimization of an autonomous mobile robot path planning based on improved genetic algorithms. *Journal of Robotics and Control (JRC)*, 4(4), 557-571. https://doi.org/10.18196/jrc.v4i4.19306

[15] Zhang, D., Luo, R., Yin, Y., & Zou, S.L. (2023). Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A∗. *Nuclear Engineering and Technology*, 55(5), 1838-1854. https://doi.org/10.1016/j.net.2023.02.005

[16] Huang, J., Zhou, B., Fan, Z., Zhu, Y., Jie, Y., Li, L., & Cheng, H. (2023). FAEL: Fast autonomous exploration for large-scale environments with a mobile robot. I*EEE Robotics and Automation Letters*, 8(3), 1667-1674. https://doi.org/10.1109/LRA.2023.3236573

[17] Morin, M., Abi-Zeid, I., & Quimper, C.G. (2023). Ant colony optimization for path planning in search and rescue operations. *European Journal of Operational Research*, 305(1), 53-63. https://doi.org/10.1016/j.ejor.2022.06.019

[18] Luan, P.G., & Thinh, N.T. (2023). Hybrid genetic algorithm based smooth global-path planning for a mobile robot. *Mechanics Based Design of Structures and Machines*, 51(3), 1758-1774. https://doi.org/10.1080/15397734.2021.1876569

[19] Preethi, P., & Mamatha, H.R. (2023). Region-based convolutional neural network for segmenting text in epigraphical images. *Artificial Intelligence and Applications*, 1(2), 103-111. https://doi.org/10.47852/bonviewAIA2202293

[20] Hebbi, C., & Mamatha, H. (2023). Comprehensive dataset building and recognition of isolated handwritten kannada characters using machine learning models. *Artificial Intelligence and Applications*, 1(3), 179-190. https://doi.org/10.47852/bonviewAIA3202624