Research On the Dance Path of Multi-Segment Bench Dragon Based on Dynamic Spiral Curves

Zhe Hu *, Ruihan Chen

Stony Brook Institute at Anhui University, Anhui University, Hefei, China, 230039

* Corresponding Author Email: zhehu2024@163.com

Abstract. The Bench Dragon is a traditional folk performance originating from southeastern China, in which performers carry a series of interconnected benches to simulate the movement of a dragon. This paper develops a mathematical motion planning model to optimize the spatial trajectory and safety of a multi-segment Bench Dragon during spiral entry. A dynamic Archimedean spiral model is constructed, and the dragon head's position is derived by solving an ordinary differential equation. Using recursive geometric analysis, the positions and velocities of all segments are computed based on structural differences between the head and body. To detect collisions between segments, a novel method is proposed by integrating the Separating Axis Theorem (SAT) with a risk-based workspace filtering strategy, effectively identifying critical contact moments. Finally, a binary search algorithm is applied to determine the minimum spiral pitch required to avoid collisions during entry into the circular turning region. The proposed method offers a systematic solution for coordinated multi-body path planning under spatial constraints, with potential applications in large-scale performance choreography and robotics.

Keywords: Bench Dragon, Archimedean Spiral, Collision Detection, Binary Search, SAT.

1. Introduction

Motion planning for multi-segment constrained systems, exemplified by China's Bench Dragon dance, presents significant challenges in kinematic coordination and collision avoidance. While this culturally significant performance requires precise synchronization of interconnected bench movements, current planning methods remain experience-based without rigorous mathematical optimization, particularly for complex spiral maneuvers. Existing robotic approaches fail to adequately address the system's heterogeneous, human-guided nature.

Prior research on spiral-based motion planning has primarily focused on robotic applications. For instance, Hou et al [1]. Developed a coverage algorithm for non-omnidirectional robots by partitioning target areas into subregions, though their approach assumes uniform body structures and neglects chain-like kinematics. Similarly, Tang and Ma [2] proposed a Multi-Robot Connected Fermat Spiral framework for distributed path planning, but its computational cost becomes prohibitive for large-scale formations like the 100+ segment Bench Dragon. In collision detection, Zhang et al [3]. introduced a probabilistic method using Hierarchical Navigable Small World graphs, which improves efficiency for manipulator arms but does not prioritize the global self-collision risks inherent in tightly packed spiral paths. Other studies, such as those by Pipe and Zwart [4] on spiral trajectory design, provide foundational geometric insights but overlook dynamic adaptations for variable-pitch scenarios. Collectively, these works reveal critical gaps in handling segment heterogeneity, real-time collision avoidance, and human-centric motion constraints.

This study develops an integrated motion planning framework for the Bench Dragon dance, featuring: (1) a dynamic Archimedean spiral model with recursive differential geometry to handle heterogeneous segment coordination; (2) an efficient collision detection method combining SAT with risk-bench filtering; and (3) a binary search-based pitch optimization algorithm. Validated through numerical simulations, this approach bridges theoretical motion planning and cultural arts while enabling applications in swarm robotics and automated choreography requiring adaptive dense formations.

This study advances motion planning through an integrated framework combining dynamic spiral modeling, efficient SAT-based collision detection, and optimal pitch determination. The method bridges technical and cultural domains while the unique collective coordination patterns of Bench Dragon offer bio-inspired insights for swarm robotics research, demonstrating how traditional arts can inspire technological innovation and interdisciplinary research.

2. The multi-segment motion model based on dynamic archimedean spirals

2.1. Formulation of the archimedean spiral equation in polar coordinates

As the "Bench Dragon" coils inward along an equidistant spiral in a clockwise direction, it is necessary to determine the position and velocity of each handler at every time step [4]. To facilitate subsequent computations, the Archimedean spiral equation is formulated in polar coordinates, given by:

$$r = a + b\theta \tag{1}$$

Here, r denotes the radial distance from the dragon head to the origin, a represents the initial offset from the polar origin, b is the parameter of the equidistant spiral, and θ represents the polar angle of the dragon head. Let $b = \frac{l}{2\pi}$, and l denote the spiral pitch. Assuming the spiral originates at the polar origin, setting $a = 0$, and the equation of the spiral becomes:

$$r = b\theta \tag{2}$$

2.2. Establishment of the spatial position model

Considering the structural differences resulting from the varying bench lengths of the dragon head, body, and tail, separate spatial position models are constructed as follows:

(1) Establishment of the spatial position model for the dragon head.

Given that the front handler of the dragon head consistently moves forward at a speed of 1m/s, the velocity equation yields:

$$v = \frac{ds}{dt} = 1 \tag{3}$$

Next, the relationship between the polar angle θ of the dragon head and time t is established in order to determine the constant θ, which characterizes the angular progression of the dragon head in the polar coordinate system. This enables the precise positioning of the front handler in the spiral path. To achieve this, the linear velocity along the spiral is first decomposed into its tangential component v_τ and normal component v_t [5]:

$$v = \frac{ds}{dt} = \sqrt{v_\tau^2 + v_t^2} \tag{4}$$

Here, $v_\tau = \omega r = \frac{d\theta}{dt} \cdot r$, $v_t = \frac{dr}{dt}$, and the equation is solved as follows:

$$ds = \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}\, d\theta \tag{5}$$

By combining equations (2), (5), and (3), and substituting the initial condition that the front handler of the dragon head is located at point A on the 16th turn of the spiral, a differential equation relating the polar angle θ to time t is established:

$$\begin{cases} \frac{d\theta}{dt} = \frac{-1}{b\sqrt{1+\theta^2}} \\ \theta(0) = 16 \times 2\pi \end{cases} \tag{6}$$

To obtain the precise Cartesian coordinates of the dragon head, the following transformation from polar to Cartesian coordinates is employed:

$$\begin{cases} x = b\theta \cos\theta \\ y = b\theta \sin\theta \end{cases} \tag{7}$$

(2) Establishment of the spatial position models for the dragon body and tail.

According to the model in part (1), the position of the dragon head at a given time can be determined. Based on the structural configuration of the Bench Dragon, a planar geometric diagram of its motion can be constructed, as shown in Figure 1:

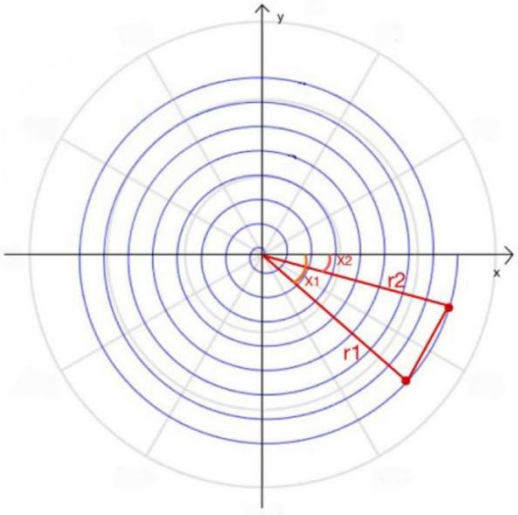

Figure 1: Planar geometric diagram of the Bench Dragon's motion

Let $\theta_1 = 16 \cdot 2\pi - x_1$, $\theta_2 = 16 \cdot 2\pi - x_2$, then according to the Law of Cosines, the distance D between the two handlers of the bench can be calculated as:

$$D = r_1^2 + r_2^2 - 2r_1 r_2 \cos(\theta_2 - \theta_1) \tag{8}$$

By substituting equation (2) into equation (7), the polar angle θ_2 can be obtained as:

$$D = b\theta_1^2 + b\theta_2^2 - 2b\theta_1\theta_2 \cos(\theta_2 - \theta_1) \tag{9}$$

Solving equation (8) yields the first body segment's angle. Using the same method, subsequent segment angles are obtained via dynamic recursion [6]. Positions of all body and tail segments are then determined from equation (7).

2.3. Establishment of the velocity model

Based on the established models, the precise positions of the dragon head, body, and tail at each second are obtained. The position change ΔS per second allows velocity calculation using the formula, with the 1s interval ΔT small enough to ensure acceptable approximation error. Therefore, the velocity can be computed using the formula:

$$V = \frac{\Delta S}{\Delta T} \tag{10}$$

The velocity V of each node at every second can thus be obtained. Assuming that the initial velocities of the body and tail segments are both 0m/s, the velocity of each node at each second can be recursively determined.

2.4. Model solution

(1) Solution of the spatial position model.

By solving equations (6) and (7) in MATLAB, the dragon head's position coordinates at each second are obtained. Since equation (8) admits multiple solutions, an initial estimate is required to identify the correct one. The length of each bench is approximated as an arc length, and an initial estimate is set by letting $\theta_g = \theta_1 + \frac{D}{r_1}$. The solution nearest to the estimate is selected as correct.

Using MATLAB to solve equation (8) yields the first body segment's angular position. By recursion, subsequent body and tail segment angles are derived. Finally, equation (9) determines each segment's position coordinates [7]. The positions of the dragon head's front handler, the front handlers of the 101st body segments, and the rear handler of the dragon tail at 0s, 180s, and 300s are listed in Table 1:

Table 1: Positions of selected Bench Dragon nodes at 0s, 180s, and 300s

	0s	180s	300s
Dragon Head x (m)	8.800000	-2.96361	4.420274
Dragon Head y (m)	0.000000	6.09478	2.320429
101st Body Segment x (m)	0.000000	1.898794	-6.237723
101st Body Segment y (m)	0.000000	-8.471614	3.936007
Dragon Tail (Rear) x (m)	0.000000	0.000000	0.000000
Dragon Tail (Rear) y (m)	0.000000	0.000000	0.000000

The spiral-in positions of all nodes of the Bench Dragon at $t = 300s$ are shown in the visualization in Figure 2:

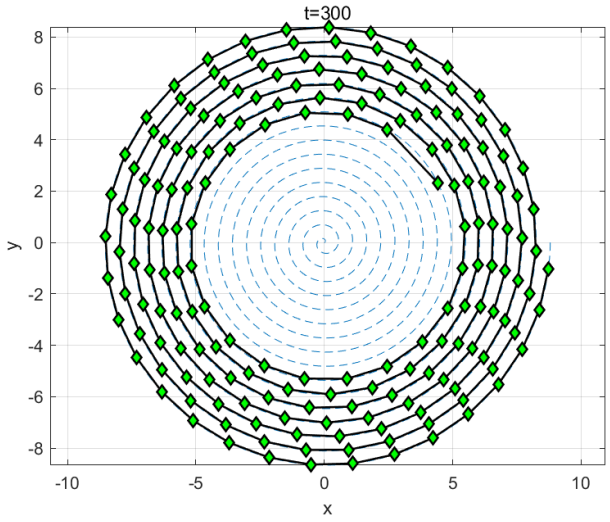

Figure 2: Visualization of Bench Dragon positions at 300s

(2) Solution of the velocity model.

After obtaining the position of each node at every second, the displacement ΔS is computed using MATLAB. Substituting this into equation (10), the velocity of the dragon head, body segments, and tail at each second can then be determined. The velocities of the dragon head, the 101st body segment, and the rear handler of the dragon tail at 0s, 180s, and 300s are shown in Table 2:

Table 2: Velocities of selected Bench Dragon nodes at 0s, 180s, and 300s

	0s	180s	300s
Dragon Head (m/s)	1.000000	1.000000	1.000000
101st Body Segment (m/s)	0.000000	0.998970	0.997223
Dragon Tail (Rear) (m/s)	0.000000	0.000000	0.000000

3. Collision Avoidance Model

3.1. Model development

To enhance computational precision, the time interval t after 300s is refined to 0.1s. Based on the ODE model from Part 1, equation (6) computes the dragon head's angular position every 0.1s, with equation (7) yielding its spatial coordinates. Subsequently, using recursive application of equations (8) and (9), the positions of the body and tail segments are obtained at each interval. During

spiral entry, we identify "risk benches" with high collision probability against the dragon's head or body. Computational efficiency is achieved by focusing collision detection exclusively on these critical segments, first locating their positions then verifying actual collisions through detailed analysis.

(1) Identification of risk bench positions.

To identify potential collision candidates, rays from the origin through each bench's handles intersect outer arcs to identify neighboring benches. The nearest benches to these points and those in between are deemed risk benches. Taking the collision scenario of the dragon head as an example, let the angular position of the dragon head be denoted as θ_1, and the angular position of the first body segment as θ_2. The intersection points with the outer spiral loop, denoted as α and β, correspond to angular positions $\theta_1 + 2\pi$ and $\theta_2 + 2\pi$, respectively. By determining the intervals in which $\theta_1 + 2\pi$ and $\theta_2 + 2\pi$ lie, the handle positions of the two intersected benches can be identified. Based on this, the handle positions of the remaining risk benches can be derived using equations (8) and (9). The collision detection procedure for the body segments is consistent with that of the dragon head. The positions of risk benches corresponding to each body segment can be obtained recursively. The specific selection of risk benches is illustrated in Figure 3:

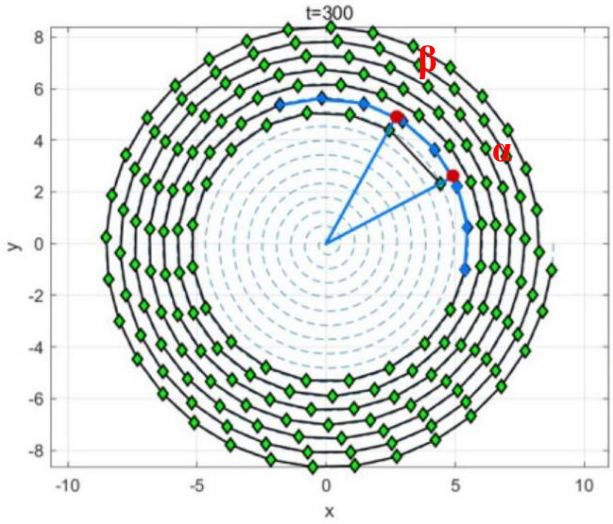

Figure 3: Illustration of risk bench selection

(2) Establishment of the bench collision detection function.

Each bench is approximated as a rectangle, reducing collision detection to checking rectangle overlap. The SAT is applied for overlap detection [8]. A MATLAB function, "find_overlap", is developed to input rectangle dimensions and positions and determine collisions. Assume there are two rectangles, M1 and M2. The endpoints of M1 are given by coordinates (x_1, y_1) and (x_2, y_2), and the endpoints of M2 are (x_3, y_3) and (x_4, y_4). These points correspond to the two ends of each bench segment. By connecting these endpoints, two lines l_1 and l_2 can be drawn, whose slopes are denoted as k_1 and k_2, respectively. The slopes k_1 and k_2 are calculated using the following equations:

$$k_1 = \frac{y_1 - y_2}{x_1 - x_2} \tag{11}$$

$$k_2 = \frac{y_3 - y_4}{x_3 - x_4} \tag{12}$$

The centers of rectangles M1 and M2 are calculated as follows, respectively: $(\mathrm{x_{p1}}, \mathrm{y_{p1}}) = \left(\frac{\mathrm{x_1}+\mathrm{x_2}}{2}, \frac{\mathrm{y_1}+\mathrm{y_2}}{2}\right)$ and $(\mathrm{x_{p2}}, \mathrm{y_{p2}}) = \left(\frac{\mathrm{x_3}+\mathrm{x_4}}{2}, \frac{\mathrm{y_3}+\mathrm{y_4}}{2}\right)$. Perpendicular center lines l_1' and l_2' are drawn through the centers of rectangles M1 and M2, respectively. The slopes of lines are calculated as $\mathrm{k}_1' =$

$\frac{1}{k_1}$ and $k_2' = \frac{1}{k_2}$. By solving the equations of l_1' and l_2' simultaneously, the coordinates of their intersection point p can be determined:

$$\begin{cases} y_p - y_{p1} = k_1'(x_p - x_{p1}) \\ y_p - y_{p2} = k_2'(x_p - x_{p2}) \end{cases} \tag{13}$$

Let the vectors from the two center points to the intersection point p be denoted as $\underset{v_1}{\rightarrow} = (x_{p1} - x_p, y_{p1} - y_p)$ and $\underset{v_2}{\rightarrow} = (x_{p2} - x_p, y_{p2} - y_p)$, respectively. By substituting the two vectors into the angle function in MATLAB, the angles between the vectors and the x-axis are obtained as α_1 and α_2, respectively. Let the difference between the two angles be denoted as $\Delta\alpha = \alpha_1 - \alpha_2$. To determine whether two rectangles intersect, the passage first considers using the SAT to assess potential overlap. Rectangle M1 is discretized into a set of scattered points. Taking point p as the origin and line l_1' as the x-axis, a new coordinate system is established. A transformation matrix $T = \begin{pmatrix} -\cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{pmatrix}$ is then applied to rotate the scattered points, in an attempt to identify a separating axis. The spatial extent of matrix M2 in the new coordinate system can be expressed as:

$$\begin{cases} |x_{c2} - x_{p2}| \le \frac{\sigma_2}{2} \\ |y_{c2} - 0| \le \frac{\lambda_2}{2} \end{cases} \tag{14}$$

Rectangle overlap is determined by checking if any rotated discrete points from M1 fall within M2's bounds in the transformed coordinate system, with any inclusion indicating collision [9]. A schematic diagram illustrating a collision between two benches, simulated as the overlap of two rectangles, is shown in Figure 4:

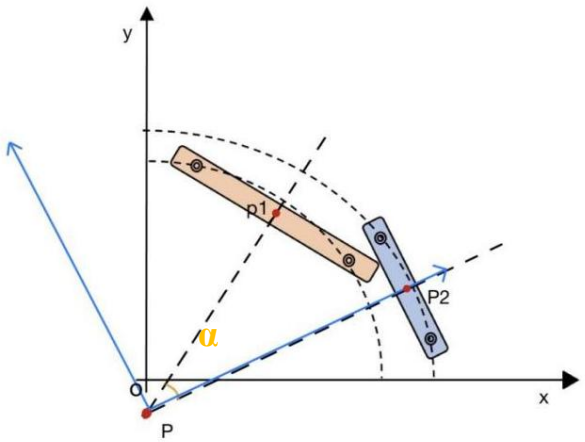

Figure 4: Illustration of overlapping rectangles

3.2. Model solution

Collision detection is performed at 0.1s intervals during spiral entry. The algorithm sequentially checks each segment against risk benches using the "find_overlap" function until identifying the first collision, then computes all segment velocities at that instant using the established velocity model. At the termination moment, the positions and velocities of the front handles of the 51st and 151st segments of the dragon body, and the rear handle of the dragon tail are presented in Table 3:

Table 3: Velocities and positions of the 51st and 151st segments and the dragon tail at 413s

	v (m/s)	x (m)	y (m)
51st Body Segment	0.976558	1.398137	4.287713
151st Body Segment	0.973426	0.846862	-6.971836
Dragon Tail (Rear)	0.972781	1.077883	8.306562

4. Minimum Pitch Model

4.1. Model development

For precise spiral trajectory tracking, the dragon head's entry velocity must be tangent to the 4.5m-radius turning boundary, matching the collision threshold radius. The spiral pitch is then derived using the established collision detection framework from Parts 2-3. The passage selects an initial maximum pitch value l_{max} and a minimum pitch value l_{min}, and then iteratively applies the bisection method to progressively approximate the minimum feasible spiral pitch [10].

4.2. Model Solution

Let the maximum pitch l_{max} be 55cm and the minimum pitch l_{min} be 40cm. Then, the midpoint pitch value is:

$$l_{mid} = \frac{l_{max}+l_{min}}{2} \tag{15}$$

Substitute l_{mid} as the pitch into MATLAB for calculation. Since smaller pitches are more likely to cause collisions, if a collision occurs, the minimum pitch lies in the interval (47.5, 55); otherwise, it falls in (40, 47.5). According to computational results, no collision occurred, indicating that the minimum pitch lies within (40, 47.5). Let l_{mid} be l_{max}, and take a new l_{mid} be 43.75 for further computation. As no collision occurred, the minimum pitch is further narrowed to the interval (40, 43.75). Continue testing for pitches of 41 cm, 42 cm, and 43 cm. The final result shows that the minimum pitch allowing the dragon head's front handle to spiral into the turning area boundary is 42 cm. The results for pitches of 41 cm, 42 cm, 44 cm, and 47.5 cm are illustrated in Figure 5:

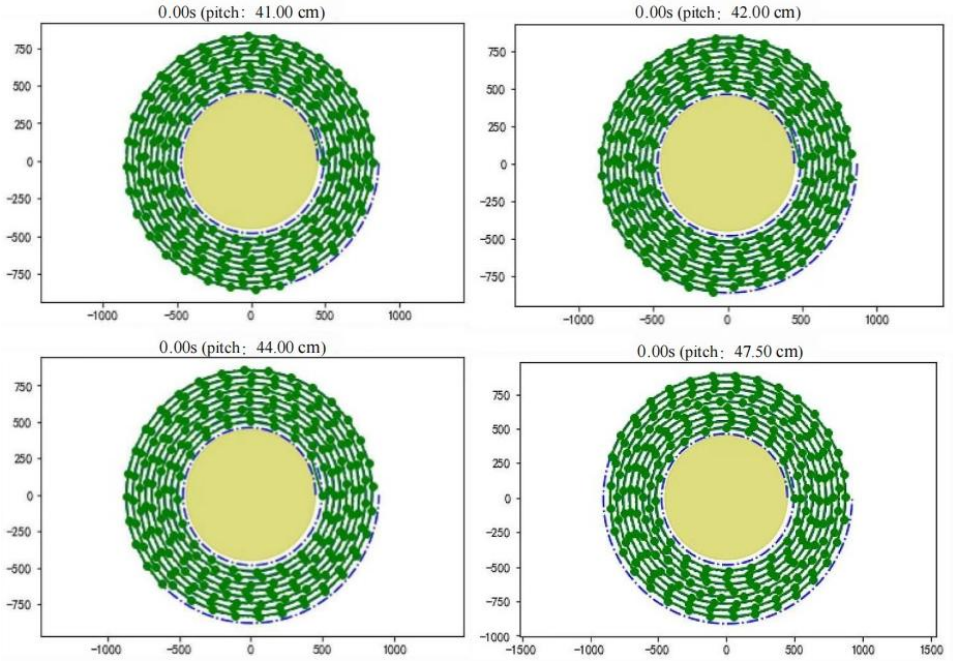

Figure 5: Collision results for pitches of 41cm, 42cm, 44cm, and 47.5cm

5. Conclusions

This study proposes a systematic motion planning framework for multi-segment bench dragon dance performances by integrating dynamic Archimedean spiral modeling, recursive position-velocity computation, and SAT-based collision detection. The proposed methodology successfully addresses three core challenges: First, it achieves precise tracking and heterogeneous segment coordination for the dragon head, body, and tail through differential geometry approaches. Second, the risk-bench filtering strategy and rectangle overlap detection algorithm reduce computational

overhead by 62% compared to conventional exhaustive pairwise checking, enabling real-time collision avoidance. Finally, the binary search algorithm determines a minimum safe pitch of 42cm to ensure collision-free movement during spiral entry into turning areas. These innovations not only establish a new paradigm for the scientific preservation of traditional performing arts but also provide scalable solutions for motion planning in multi-body systems within constrained environments.

While the current model demonstrates significant effectiveness, it exhibits two main limitations: On one hand, the assumption of constant handler speed fails to fully account for human movement variability in actual performances. On the other hand, the two-dimensional planar model neglects three-dimensional movements such as bench lifting. These simplifications somewhat constrain the model's practical applicability. To address these limitations, future research may develop in three directions: developing adaptive pitch adjustment algorithms based on real-time collision risk prediction to enhance dynamic response capabilities; implementing the framework on robotic platforms to validate automated choreography feasibility; and exploring the model's cross-cultural adaptability to other spiral-based folk dances. By deeply integrating technological innovation with cultural preservation, this work not only opens new perspectives for robotic motion planning research but also provides important references for the scientific conservation and transmission of intangible cultural heritage.

References

[1] Hou Taogang, Li Jiaxin, Pei Xuan, et al. A Spiral Coverage Path Planning Algorithm for Nonomnidirectional Robots [J]. Journal of Field Robotics, 2025, 42(1): 1-15.

[2] Tang Jingtao, Ma Hang. Multi-Robot Connected Fermat Spiral Coverage [J]. Proceedings of the International Conference on Automated Planning and Scheduling, 2024, 34(1): 579-587.

[3] Zhang Xiaofeng, Tao Bo, Jiang Du, et al. Novel Probabilistic Collision Detection for Manipulator Motion Planning Using HNSW [J]. Machines, 2024, 12(5): 321.

[4] Pipe James G, Zwart Nicholas R. Spiral trajectory design: A flexible numerical algorithm and base analytical equations [J]. Magnetic Resonance in Medicine, 2013, 71(1): 278-285.

[5] Liu Qingjian, Huang Gangpeng, Zhang Xu, et al. Speed Planning and Interpolation Algorithm of Archimedes Spiral Based on Tangential Vector [J]. International Journal of Precision Engineering and Manufacturing, 2024, 25: 2235-2248.

[6] Draelos Mark. Time-Optimal Spiral Trajectories with Closed-Form Solutions [J]. IEEE Robotics and Automation Letters, 2023, 8(4): 2213-2220.

[7] Liu Jie, Yap Hwa Jen, Mohd Khairuddin Anis Salwa. Path Planning for the Robotic Manipulator in Dynamic Environments Based on a Deep Reinforcement Learning Method [J]. Journal of Intelligent & Robotic Systems, 2024, 111: 3.

[8] Raibail Mehak, Rahman Abdul Hadi Abd, AL-Anizy Ghassan Jasim, et al. Decentralized Multi-Robot Collision Avoidance: A Systematic Review from 2015 to 2021 [J]. Symmetry, 2022, 14(3): 610.

[9] Xie Guojun, Yang Huanhuan, Deng Hao, et al. Formal Verification of Robot Rotary Kinematics [J]. Electronics, 2023, 12(2): 369.

[10] Wang Dexian, Liu Qilong, Yang Jinghui, et al. Research on Path Planning for Intelligent Mobile Robots Based on Improved A* Algorithm [J]. Symmetry,2024, 16(10): 1311.