Curve Architecture Optimization Based on Multiple Search Algorithms

Huipeng Xie [†, *], Zhengbin Xu [†] and Fengyuan Hu [†]

School of Mathematical Sciences, Chengdu University of Technology, Yibin, China

[†] These authors also contributed equally to this work

* Corresponding Author Email: 3407751918@qq.com

Abstract. The purpose of this paper is to construct and analyze a model for collision detection and velocity regulation in curved architectures based on multiple search algorithms using curve function and differential equation models. The study first calculates the arc length by curve equations and first-class curve integrals to determine the position and velocity of the curve architecture per second. Based on this, the differential equations of the polar angles of each part of the curved architecture are modeled with respect to time, and the approximate solutions are obtained using the Runge-Kutta method. Then, algebraic and geometric knowledge is applied to construct the position equations. In addition, this paper defines the collision conditions and constructs a collision detection model by an iterative method to finally derive the termination time. Finally, the relationship between the inverse curvature distance and the Euclidean distance between the parts of the curve architecture is demonstrated for the speed control aspect, and the multiple search algorithm is used to perform an accurate search and derive the minimum curvature distance corresponding to the maximum Euclidean distance. The results of this paper provide an effective optimization scheme for curved architectures, which is especially important in areas such as machine motion planning.

Keywords: Curve architectures, Runge-Kutta method, differential equations, multiple search algorithms.

1. Introduction

In this paper, the Runge-Kutta method [1] and differential equations [2] are comprehensively applied, while multiple search algorithms [3] are used to analyze the motion state of the curved architecture [4], and the dynamic characteristics of the curved architecture during the spiral process are deeply investigated. Firstly, the motion trajectory of the curve architecture in the specified time is calculated by setting the initial conditions. Secondly, focusing on detecting the interactions between the sections, the termination moment of the curve disking in was determined and the final position and velocity were recorded. Finally, the optimization of the minimum curvature pitch in the turnaround space [5] was explored to ensure that the head of the curve architecture can enter the turnaround region smoothly. This study not only provides a new perspective for the dynamic analysis of curved architectures, but also provides a useful reference for algorithmic applications in the field of machine motion [6].

2. Curve Architecture Position and Velocity Analysis

The goal of this section is to calculate the position and velocity of the entire curved architecture per second. The parameters of the section are known, and the curve architecture coils clockwise along an equally spaced curve with a curvature of 55 cm, with the center of each handle evenly spaced across the curve. In cases where the analytical velocity solution is difficult to obtain, numerical differentiation methods can be used, such as difference quotient type numerical differentiation formulas versus interpolation type numerical differentiation formulas. It should be noted that the velocity is a vector quantity, which requires not only the magnitude of the velocity of the curve architecture at each moment, but also the direction of its velocity at some moments.

2.1. Curve Architecture Modeling

The polar coordinate equations of an isometric curve are expressed as follows:

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

r is the polar diameter in polar coordinates, θ is the polar angle, and a,b are all parameters. Suppose the initial angle of the curve at the origin is 0. Then $b=0$ .By the special property of isometric curves, for any two polar angles α,β, when these two polar angles satisfy the following relationship:

$$\alpha = \beta + 2\pi \tag{2}$$

Their corresponding polar angles are satisfied:

$$r_\alpha - r_\beta = p \tag{3}$$

Where p is the curvilinear distance. The associative formula is obtained:

$$a = \frac{p}{2\pi} \tag{4}$$

Further, the equidistant curve equation is obtained as:

$$r = \frac{p}{2\pi}\theta \tag{5}$$

Transform each point (r,θ) on the curve into coordinates under the Cartesian coordinate system (x,y).

$$\begin{cases} x = a\theta\cos\theta \\ y = a\theta\sin\theta \end{cases} \tag{6}$$

Assume that initially, the center of the front handle of the head of the curve architecture is at point A. Let the front handle of the head of the curved structure be the first handle, and then $2,3,\cdots,\ 224$ handle. Let the position of the i rd handle be (x_i,y_i) , and the polar angle of the i rd handle be θ_i , then the polar angle of the center of the front handle of the head of the curved architecture θ_1 , and the polar angle of the front handle of the head of the curved architecture at the initial moment is θ_0 . Known. Let the action distance of the front handle of the head of the curved architecture be s , the action time be t , and the action speed be v. Then:

$$s = vt \tag{7}$$

On the other hand, by integrating the curves of the first type, one obtains.

$$s = \int_{\theta_0}^{\theta_1} \sqrt{(a\cos\theta_1 - a\theta_1\sin\theta_1)^2 + (a\sin\theta_1 + a\theta_1\cos\theta_1)^2}\, d\theta_1 \tag{8}$$

The travel speed of the front handle of the curved architecture head is always 1 m/s, so the action speed of the front handle of the curved architecture head is numerically equal to 1. Hence the equation is obtained:

$$\int_{\theta_0}^{\theta_1} \sqrt{(a\cos\theta_1 - a\theta_1\sin\theta_1)^2 + (a\sin\theta_1 + a\theta_1\cos\theta_1)^2}\, d\theta_1 = t \tag{9}$$

The equation is derived from θ_1 on both sides at the same time, and then the reciprocal is taken on both sides at the same time, and it is noted that θ_1 varies from θ_0 downwards, so that we get:

$$\frac{d\theta_1}{dt} = -\left(\sqrt{(a\cos\theta_1 - a\theta_1\sin\theta_1)^2 + (a\sin\theta_1 + a\theta_1\cos\theta_1)^2}\right)^{-1} \tag{10}$$

It is possible to obtain $\theta_1=\theta_1(t)$ from Eq. Obviously, at the initial moment, $\theta_1(0)=\theta_0$. This gives the position of the head of the curve architecture per second:

$$\begin{cases} x_1=a\theta_1(t)\cos\theta_1(t) \\ y_1=a\theta_1(t)\sin\theta_1(t) \end{cases} \tag{11}$$

Let the magnitude of the velocity of the i nd handle be v_i , and its components along the x,y -axis direction be v_{ix},v_{iy} , respectively, and the derivation of Eq. with respect to time t yields the magnitude of the velocity of the handle in front of the head of the curved architecture as:

$$\begin{cases} v_{1x}(t)=\dfrac{dx_1(t)}{dt} \\ v_{1y}(t)=\dfrac{dy_1(t)}{dt} \\ v_1(t)=\sqrt{v_{1x}^2+v_{1y}^2} \end{cases} \tag{12}$$

From this paper, it is easy to know that the distance between the centers of the first two handles and the centers of the remaining handles are constant, let the distance between the center of the i nd handle and the center of the $i+1$ rd handle is $L_i, i=1,2,\cdots,223$, it is easy to know that.

$$L_2=L_3=\cdots=L_{223} \tag{13}$$

So let $L_2=L_3=\cdots=L_{223}=L$, then the position of the $i+1$ th handle can be calculated by the following equation.

$$\begin{cases} \sqrt{(x_i-x_{i+1})^2+(y_i-y_{i+1})^2}=L_1, i=1 \\ \sqrt{(x_i-x_{i+1})^2+(y_i-y_{i+1})^2}=L, else \end{cases} \tag{14}$$

This gives the position of the $i+1$ nd handle as $(x_{i+1}(t),y_{i+1}(t))$. Similarly, its derivative with respect to time t yields the velocity of the $i+1$ th handle as.

$$\begin{cases} v_{i+1x}(t)=\dfrac{dx_{i+1}(t)}{dt} \\ v_{i+1y}(t)=\dfrac{dy_{i+1}(t)}{dt} \\ v_{i+1}(t)=\sqrt{{v_{i+1x}}^2+{v_{i+1y}}^2} \end{cases} \tag{15}$$

Finally, the mathematical model for this section is developed:

$$\begin{cases} C_1 = \dfrac{p}{2\pi}\cos\theta_1(t) - \dfrac{p}{2\pi}\theta_1(t)\sin\theta_1(t) \\ C_2 = \dfrac{p}{2\pi}\sin\theta_1(t) + \dfrac{p}{2\pi}\theta_1(t)\cos\theta_1(t) \\ \dfrac{d\theta_1(t)}{dt} = -\left({C_1}^2 + {C_2}^2\right)^{-\frac{1}{2}} \\ x_1(t) = \dfrac{p}{2\pi}\theta_1(t)\cos\theta_1(t) \\ y_1(t) = \dfrac{p}{2\pi}\theta_1(t)\sin\theta_1(t) \\ v_1(t) = \sqrt{\left(\dfrac{dx_1(t)}{dt}\right)^2 + \left(\dfrac{dy_1(t)}{dt}\right)^2} \\ L_i = \sqrt{(x_i(t) - x_{i+1}(t))^2 + (y_i(t) - y_{i+1}(t))^2} \\ v_{i+1}(t) = \sqrt{\left(\dfrac{dx_{i+1}(t)}{dt}\right)^2 + \left(\dfrac{dy_{i+1}(t)}{dt}\right)^2} \\ i = 1, 2, 3, \cdots, 223 \end{cases} \tag{16}$$

2.2. Solving for Position and Velocity

For the model that has been developed in this section, it is first necessary to solve the first order ordinary differential equation.

$$\begin{cases} \dfrac{d\theta_1(t)}{dt} = f(t, \theta_1), 0 \leqslant t \leqslant 300 \\ \theta_1(0) = \theta_0 \end{cases} \tag{17}$$

Considering that the differential equations cannot be solved analytically, it is decided in this paper to approximate them numerically with the help of the Runge-Kutta method. The fourth order Runge-Kutta method is chosen to solve the numerical solution of this differential equation. The fourth-order standard Runge-Kutta method, as a high-precision single-step explicit method, has a simple computational procedure and can meet the accuracy requirements. In this section, the fourth-order standard Runge-Kutta formulation is as follows:

$$\begin{cases} \theta_{1n+1} = \theta_{1n} + \dfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(t_n, \theta_{1n}) \\ K_2 = f\left(t_n + \dfrac{1}{2}h, \theta_{1n} + \dfrac{1}{2}hK_1\right) \\ K_3 = f\left(t_n + \dfrac{1}{2}h, \theta_{1n} + \dfrac{1}{2}hK_2\right) \\ K_4 = f(t_n + h, \theta_{1n} + hK_3) \end{cases} \tag{18}$$

Where h is the step size and the local truncation error is $\theta_1(t_{n+1}) - \theta_{1n+1} = O(h^5)$.

The next step in solving numerical differentiation involves approximating the derivative value of a function at a node using the function value at the discrete node. The three-point formula is an ingenious method used to solve numerical differentiation by interpolating at specific points to approximate the derivative. In this section, for a given function value $x_i(t_0), x_i(t_1), x_i(t_2)$ on three nodes $t_0, t_1 = t_0 + h, t_2 = t_0 + 2h$, there are:

$$\begin{cases} v_i(t_0) = x_i{}'(t_0) = \dfrac{1}{2h}\left[-3x_i(t_0) + 4x_i(t_1) - x_i(t_2)\right] \\ v_i(t_1) = x_i{}'(t_1) = \dfrac{1}{2h}\left[-x_i(t_0) + x_i(t_2)\right] \\ v_i(t_2) = x_i{}'(t_2) = \dfrac{1}{2h}\left[x_i(t_0) - 4x_i(t_1) + 3x_i(t_2)\right] \end{cases} \tag{19}$$

The known parameters provided in the text: $p = 0.55m, L_1 = 2.86m, L = 1.65m$ are substituted into the formula and the Runge-Kutta method is applied to solve the differential equations involved in the formula. Also, the three-point formula in the interpolated numerical differential equation is utilized to perform the numerical differential approximation involved in Eq.

The positions and velocities of the front handle of the head of the curve architecture, the front handle of the 1st, 51st, 101st, 151st, and 201st sections of the body behind the head of the curve architecture, and the rear handle of the tail of the curve architecture are solved for 0s, 60s, 120s, 180s, 240s, and 300s, and from the results, it can be seen that the positions and velocities of the different sections change as the movement of the curve architecture proceeds. For example, the position of the head of the curve architecture along the x-axis and y-axis changes considerably at different points in time, while the velocity remains relatively constant. From the head of the curved architecture to the tail of the curved architecture, the velocity value of each node shows a gradual decrease, which may be due to the attenuation phenomenon in the process of energy transfer during the action of the curved architecture.

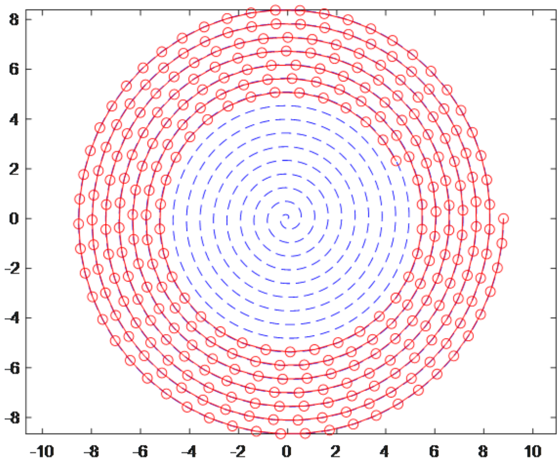

Fig. 1 Motion trajectory of the handle in front of the head of the curved architecture

Fig. 1 presents the trajectory of the handle in front of the head of the curved architecture, which contains two curves: one is the blue dashed line, and the other is the red solid line. The blue dashed line represents the disk-in curve, while the red solid line represents the trajectory of the handle in front of the head of the curved architecture. In the trajectory of the curved head handle, the distance between any two neighboring positions is nearly the same, which is due to the constant velocity of the curved head handle.

3. Collision Detection Analysis

This section extends the simulation of the curve architecture motion path based on the previous section until the simulation is terminated when a collision occurs. The first task is to construct a collision detection model, which does abstraction of the knots and defines knots collision as intersections on the plane. The collision detection model is built by detecting whether the knots intersect in the plane or not. Over time, until the collision occurs, the collision moment and its neighboring time (excluding future moments) are considered as the termination point, and the previous model is used to calculate the position and velocity of the curve architecture.

3.1. Collision Modeling

The entire curve architecture is carefully analyzed, and the collision event must first affect the head of the curve architecture. In the isometric curve scenario, the space in which the head of the curve moves toward the origin is narrower, and the probability of collision increases accordingly. In fact, the probability of collision is higher for the head of the curve than for the other sections. Therefore, it is reasonable to perform collision detection only on the head of the curved architecture to determine the termination moment. Abstracting the knots as thickness-free rectangles, a sufficient condition for two knots to collide is that their rectangular regions overlap. Note that rectangle overlap is equivalent to one rectangle vertex being inside the other rectangle. Therefore, it is only necessary to check whether any vertex of the section at the head of the curve architecture is inside the rectangular region of the other section (except for the first section body). If the vertex is inside the section, it is determined to be a collision; otherwise, there is no collision.

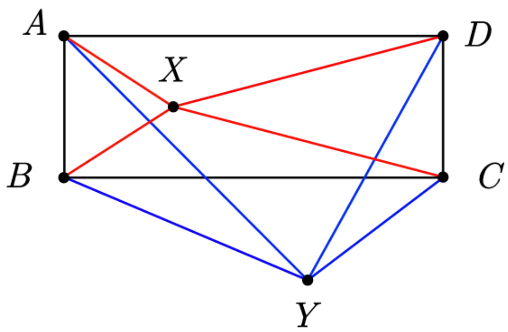

Fig. 2 Points inside and outside the rectangle

As shown in Fig. 2, rectangle $ABCD$, point X is inside the rectangle and point Y is outside the rectangle. Let for any point Z connected with the four vertices of the rectangle, respectively, from four triangles: $\triangle ZAB, \triangle ZAD, \triangle ZCD, \triangle ZBC$, the corresponding areas are denoted as: S_1, S_2, S_3, S_4. Then, from Fig. 2, it follows that:

$$\begin{cases} \sum_{i=1}^{4} S_i = S_{ABCD}, & \textit{Point inside the rectangle} \\ \sum_{i=1}^{4} S_i > S_{ABCD}, & \textit{Point outside the rectangle} \end{cases} \tag{20}$$

Analytically, when a vertex of the head of the curve architecture satisfies the formula inside the rectangular area (e.g., in Fig. 2, the position of point X), the sections collide; otherwise (e.g., in Fig. 2, the position of point Y), no collision occurs.

Let the head of the curve architecture be section 1, followed by section $2, 3, \cdots, 223$, $S_{ij}(k)$ which is the j th area of the k th vertex of the head of the curve architecture with respect to the i th section, and the length of the i th section is l_i and the width w. Thus, the collision condition is:

$$\sum_{j=1}^{4} S_{ij}(k) = l_i w, i = 3, \cdots, 223, \quad k = 1, 2, 3, 4 \tag{21}$$

However, there is a rounding error, so it is corrected to a collision detection model:

$$\sum_{j=1}^{4} S_{ij}(k) \leqslant l_i w, i = 3, \cdots, 223, \quad k = 1, 2, 3, 4 \tag{22}$$

3.2. Collision Model Solution

Using the Helen's formula and solving first with a time step of 1s, the result of 414s is obtained, which means that the curve architecture has a collision time in interval $(413, 414]$. To obtain a more accurate result, searching with 413s as the starting point and a step size of 0.01s, the following result is obtained as in Fig. 3.

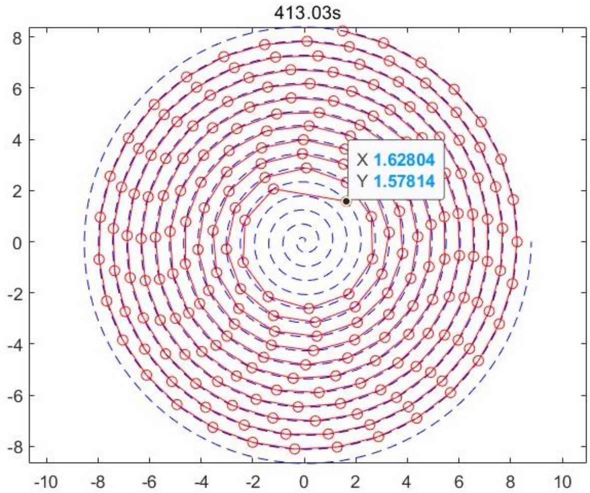

Fig. 3 The collision moment and the motion trajectory of the curve architecture

It is stated that the collision time of the curve architecture occurs at 413.03s, so the termination moment can be 413.02s. In other words, no collision occurs between the nodes (i.e., the curve architecture cannot continue to be coiled in) at $t = 413.02s$.

Finally, with t=413.02s known, the position and velocity of the curve architecture at this point are solved using Eq. Partial results are shown in table 1.

Table 1. Partial results of the position and velocity of the curve architecture now of termination

Structure	Position	Speed (m/s)
Curved architecture head x (m)	1.621347	1.000000
Curve architecture head y (m)	1.585569	
Section 1 x (m)	-1.199008	0.991156
Section 1 y (m)	2.060124	
Section 51 x (m)	1.778852	0.975967
Section 51 y (m)	4.135598	
Section 101 x (m)	-1.062092	0.974345
Section 101 y (m)	-5.800207	
Section 151 x (m)	0.43945	0.972425
Section 151 y (m)	-7.004213	
Section 201 x (m)	-7.951723	0.970989
Section 201 y (m)	-0.702747	

4. Minimum Curve Pitch Exploration

A simple analysis shows that since each section is a rectangle with a width, the smaller the curvature p the more likely it is that an inter-section collision will occur, which corresponds to an inevitably smaller running time, and implies that the distance between the front handle of the head of the curvilinear architecture and the origin d is larger. It follows that the curvature p is positively correlated with $\frac{1}{d}$.

In this section, it is required to find a minimum curvature distance such that the curve architecture reaches the boundary of the turnaround space with the front handle of the head of the curve architecture without collision. Then it is easy to find a critical condition: the corresponding curvature distance p is the minimum curvature distance when the derived d in the case of a collision reaches its maximum.

4.1. Minimum Curve Pitch Modeling

Let the radius of the boundary circle be R, when $p=0.55m$, it is easy to know that $d=1.62804^2+1.57814^2=5.14104$, the unit m. From this paper, $d_{\max}=R$. Therefore, it is only necessary to search the approximate range of p starting from $p_0=0.55$ with a fixed step size, and then take a smaller step size for grid search near the optimal result, so as to obtain more accurate results.

4.2. Minimum Curvature Pitch Model Solving

(1) Preliminary search

Take 0.001 meter as the step size, search for the curvature distance in the range, and substitute it into the collision detection model until a collision occurs. Record the position of the handle in front of the head of the curve architecture at this point and calculate the Euclidean distance between this position and the origin. If this distance exceeds the radius of the header space, the next iteration is performed. Eventually, it is concluded that the minimum curvilinear distance lies between (0.450, 0.451) meters, from which the second round of search is initiated.

(2) Accuracy search

For the second search, the smallest curvature distance is searched within (0.450, 0.451) m with a step size of 0.000001 m. The above Euclidean distance is minimized when the smallest curvature distance is obtained to be 0.450343m. The solving result section is shown in Fig. 4.

To solve the model in this section, the curves shown in Fig. 4 are obtained by first searching the grid in steps of $h = 0.001m$.

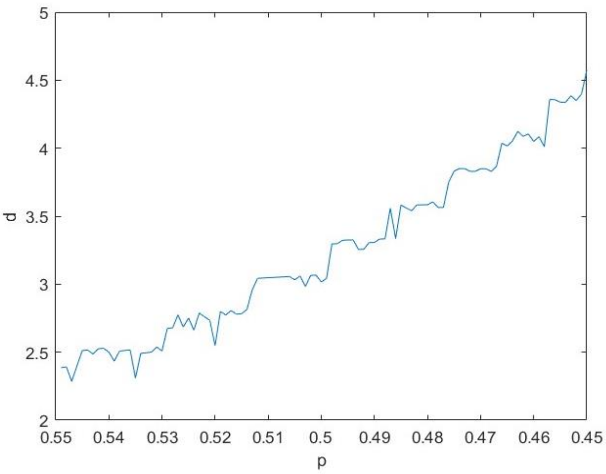

Fig. 4 Curve versus distance searched in steps of 0.001m

Fig. 4 illustrates the image of the function $d(p)$, where p represents the curvature distance and d represents the Euclidean distance between the center of the front handle of the head of the curved architecture and the origin at the time of collision. The figure depicts the complex relationship between the distance between the center of the front handle of the head of the curved architecture and the origin as a function of the curvature distance. The searched results are $p = 0.45\,m$ and $d = 4.57194\,m$. This indicates that at $p = 0.45\,m$, the collision has already occurred before the curved architecture head front handle has entered the turnaround space. Therefore, it is necessary to refine the search interval to, and reduce the search step size by changing the search step size to 0.000001m after the grid search to obtain the results shown in Fig. 5.

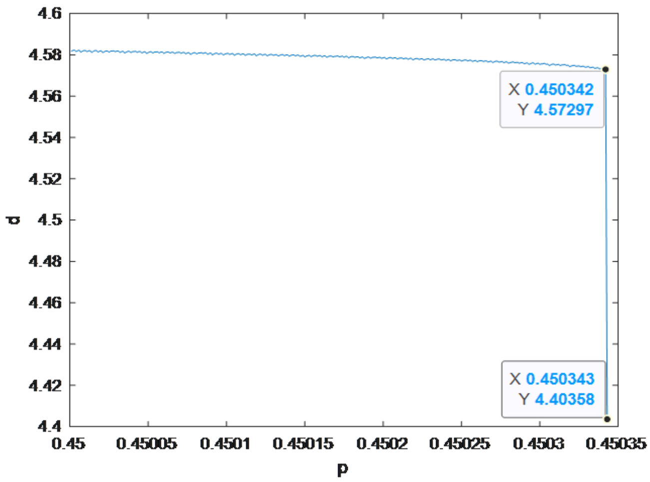

Fig. 5 Curvature versus distance searched with a step size of $0.000001m$

The Euclidean distance between the center of the front handle of the head of the curved architecture and the origin is about 4.57297 m at the curvilinear distance $p = 0.450342\ m$, while the distance is about 4.40358 m at the curvilinear distance $p = 0.450343\ m$. With the small change of the curvilinear distance of only 0.000001 m, the front handle of the head of the curved architecture can enter the head turning space smoothly without collision. Therefore, it can be inferred that $p = 0.450343m$ is the smallest curvature boundary that allows the front handle of the curved head to enter the turnaround space.

5. Conclusion

In this paper, a mathematical model based on differential equation model and multiple search algorithm is constructed for the optimization of path and velocity control scheme of curved architecture. By introducing the curve equation and the first-class curve integral, the motion trajectory of the curved architecture is effectively calculated, and the velocity and position changes of each section are found by using the Runge-Kutta method. In terms of collision detection, the interaction between the sections is defined and the safe termination time of the curve motion is determined by recognizing the collision conditions in time through an iterative method. Finally, by analyzing the inverse relationship between the Euclidean distance and the curve distance, the design of the turnaround space is successfully optimized, and the minimum curve distance is precisely solved by applying the multiple search algorithm. In summary, this research not only provides a new method for the dynamic analysis of curvilinear architectures, but also provides a reference for the development of related technologies in the field of machine motion planning and other areas.

References

[1] Wang Jingzhi, Chen Honglin. Research on parachute airdrop based on Runge-Kutta algorithm [J]. Modern Electronic Technology,2010,33(14):124-126+130.DOI: 10.16652/j.issn.1004-373x.2010.14.004.

[2] Chen Yutong, Xu Junfeng. Solutions of a class of nonchiral linear complex differential equations on a circle [J]. Journal of Wuyi University (Natural Science Edition),2024,38(04):36-41.

[3] Hu Shuli. Research on heuristic search algorithm for solving combinatorial optimization problems[D]. Northeast Normal University, 2019.DOI: 10.27011/d.cnki.gdbsu.2019.000102.

[4] Wang Hui, Zhang Tienan, Zhu Jian. Architecture design of plant and station side planned value curve receiving system [J]. Electrotechnology,2022, (22): 220-222.DOI: 10.19768/j.cnki.dgjs.2022.22.068.

[5] Wan Kai, Sun Yanpan, Wang Tao, et al. Construction technology of shield turnaround in closed subway station [J]. Jiangsu construction,2018, (05):65-67+80.

[6] Ge Yaming, Chen Jiehao. Model predictive contour control for robotic crowd motion planning [J]. Laboratory Research and Exploration,2024,43(07):28-33+58.DOI: 10.19927/j.cnki.syyt.2024.07.006.