Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 13, No. 1, 2024

Research on Spiral Motion Analysis Based on Optimisation Algorithm and Neural Networks

Weishuai Cui[1, †, *], Xi Deng[1, †], Peiting Dong[2]

[1]Naval Architecture and Ocean Engineering College, Dalian Maritime University, Dalian, China
[2]School of Maritime Economics and Management, Dalian Maritime University, Dalian, China
[†] These authors also contributed equally to this work
[*]Corresponding author: 2167731367@qq.com

Abstract: This paper presents an in-depth study of motion problems involving dynamical models in spiral paths, with a focus on computational techniques and algorithms. By utilising the geometric properties of spirals, we have developed equations of motion for the nodes along the path [1]. The dynamics of motion in the spiral path are accurately described using advanced computational methods. First, we developed a motion model based on the geometric properties of the helix. The complete motion trajectory is obtained by second-by-second enumeration using MATLAB to ensure accurate computational precision. Second, a collision detection model is constructed; using MATLAB's inpolygon function, we determine that the collision occurs at 412.5 seconds. Finally, we use advanced algorithms to optimise the motion path and velocity. A particle swarm optimisation (PSO) algorithm and a feed-forward neural network are used to determine the minimum pitch angle, which is calculated to be 43.4 cm [2,3]. In addition, the maximum velocity that satisfies the velocity constraint is solved by these computational techniques.

Keywords: Spiral Path, Collision Detection, Motion Optimization, Position Distribution.

1. Introduction

Modelling and optimisation of complex motion systems is an important research direction in modern computer science and engineering. Motion problems in spiral paths involve multiple disciplines such as geometry, dynamics and algorithmic optimisation, and are an important foundation for understanding and solving practical engineering problems. However, traditional motion models and algorithms often suffer from insufficient accuracy and inefficiency when dealing with complex paths. Therefore, it is of great theoretical and practical significance to improve motion models and optimisation algorithms using advanced computational methods and machine learning techniques [4].

The research objective of this paper is to deeply explore the motion problem in spiral paths through geometric characteristic analysis, collision detection model construction and path optimisation based on particle swarm optimisation algorithm and feed-forward neural network. With accurate simulation and calculation through MATLAB, we can not only accurately describe the motion trajectory, but also effectively avoid collision and optimise the motion speed and path [5].

The research results in this paper not only provide new solutions for motion problems in complex paths, but also demonstrate the great potential of computer science and technology in engineering applications. It is hoped that this research can provide valuable references and lessons for further research in related fields.

2. Helical Motion Modeling

2.1. Modeling

The dragon dance team coils in clockwise along an isometric screw thread with a pitch of 55 cm, and the velocity of the handle in front of the dragon head is constant at 1 m/s. Since the dragon body consists of multiple nodes, the velocity and position of the nodes change over time and show complex behaviors. For this reason, a model needs to be built to calculate and record the position and velocity of each node between 0 and 300 seconds. The motion of the dragon's head drives the entire dragon body, so the analysis needs to consider the geometric properties of the helix and the interactions between the nodes. Specifically, the motions of the nodes of the dragon body should be derived by recursive equations, which use the motion state of the dragon head as the initial value to derive the motions of each node. The model aims to accurately describe the positions and velocities of the dragon head, different nodes of the dragon body and the dragon tail manages at different moments, to comprehensively demonstrate the motion characteristics of the dragon dance team.

A helix is a curve that is constantly circling to the center and is characterized by equidistant circling, i.e., the radial distance decreases by a certain amount for each rotation around the center. Establish the coordinate system shown in the figure and describe the problem in different coordinate forms as in Figure 1.

The polar form of the helix is:

$$r(\theta)=\frac{p}{2\pi}\theta \tag{1}$$

The Cartesian coordinate form of the helix is:

$$\begin{cases} x=r(\theta)\cos\theta \\ y=r(\theta)\sin\theta \end{cases} \tag{2}$$

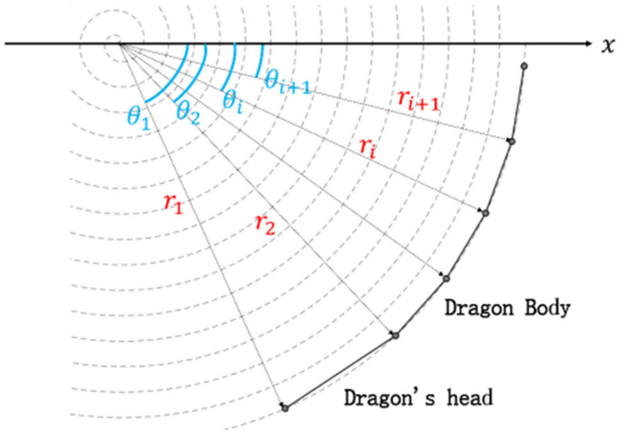

Figure 1. Schematic diagram of the position of the bench dragon

2.2. Solving the model

Providing that the helix is positively oriented counterclockwise from the center, the angle θ of the handle in front of the actual problem faucet: $32\pi \rightarrow 0$.

In polar coordinates, the displacement of the faucet handleds is:

$$ds^2 = dr^2 + (rd\theta)^2 \tag{3}$$

where ds is the microdisplacement along the helix, anddr is the polar diameter microelement, and $rd\theta$ is the polar angle microelement.

Substituting (1) into (2) gives:

$$ds = \sqrt{K^2 + K^2\theta'^2}\, d\theta \tag{4}$$

Deriving expression (4) for time t yields:

$$\frac{ds}{dt} = \sqrt{K^2 + K^2\theta'^2}\,\frac{d\theta}{dt} \tag{5}$$

The velocity of the faucet is constant at 1 m/s, i.e., $ds/dt = 1$, and the polar angle of the faucet is obtained by substituting θ into the first-order constant coefficient differential equation with respect to time t.

$$\frac{d\theta}{dt} = \frac{1}{K\sqrt{1+\theta^2}} \tag{6}$$

Among them $K = \frac{p}{2\pi}$.

After obtaining this differential equation, solve its analytical solution to get the θ with time t The exact functional relationship can be obtained. However, after trial calculations, the differential equation has no analytical solution, so only its numerical solution can be obtained.

ode45 is a general-purpose ordinary differential equation solver in MATLAB, which is a middle-order, adaptive step-size way to solve non-rigid ordinary differential equations. For equation (5), code it and use ode45 to solve this differential equation to get the dragon head pole angle θ time-dependent t. The variation curve of the faucet's front handle per second, i.e., the polar angle of the faucet's front handle per second θ_t.

Visualization of the obtained results for a more intuitive presentation θ The relationship with t the relationship is shown in Figure 2:

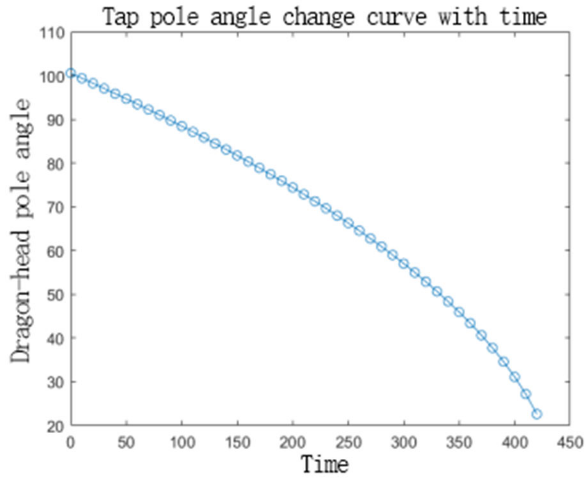

Figure 2. Polar angle of the faucet as a function of time.

Based on the cosine theorem, a recurrence relation for the angles is derived and the position of each node of the dragon's body and tail per second is calculated by substituting the angle per second of the handle in front of the dragon's head. For the speed of the coiled dragon procession, the instantaneous speed of the mass point is approximated by the difference method. The angles at different points in time are selected to determine the corresponding node positions and converted into Cartesian coordinate form. The velocities at the start, end and intermediate time points are calculated by the forward, backward and center difference methods, respectively, to solve for the velocity of the leading forehandle at any moment. This method is also suitable for recursively calculating the velocities of each node of the dragon body and dragon tail per second to ensure the coordination and efficiency of the overall movement.

Based on the positional information of the panglong team per second, the visualization of the panglong team was realized by using MATLAB, as shown in Figure 3:

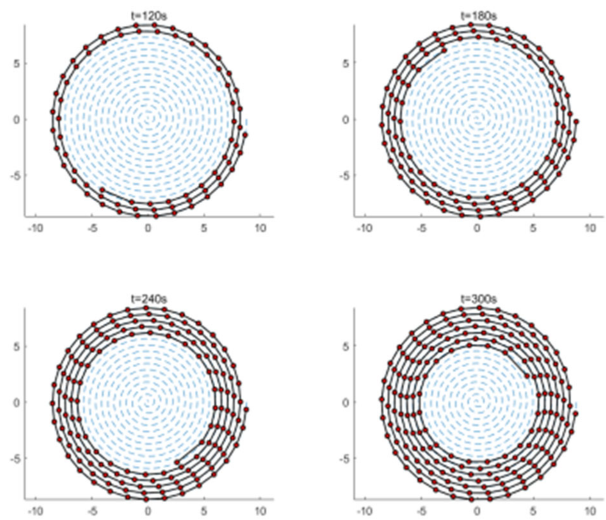

Figure 3. Position of the bench dragon at different moments

3. Calculation of the Moment of Termination of The Safe Disk Entry

3.1. Modeling

As the dragon dance team coils in along the spiral, the polar diameter gradually decreases, resulting in the nodes of the dragon body gradually approaching each other. It is necessary to determine the termination moment of the dragon team without collision during the coiling process, i.e., to find the moment when the minimum distance between the nodes reaches a critical value. For this purpose, the motion at each moment must be accurately simulated, and the length and width of the benches and the way the handles are connected must be considered to ensure that the dragon bodies do not collide with each other. Calculations are made to derive the furthest distance and time that the dragon dance team can coil in, and to provide the positions and velocities of the nodes under the termination moment. At the end of the coiling in, the dragon dance team needs to make a turnaround to coil out counterclockwise. The key is to determine the minimum pitch to ensure that the front handle of the dragon head can enter the turnaround space with a diameter of nine meters. The smaller the pitch, the tighter the coiling path and the more fully the turnaround space is utilized. The analysis required the construction of a geometric model describing the faucet's travel along the thread and into the turnaround space, which involved an in-depth understanding of the geometric properties of the thread to ensure that the turnaround process was completed successfully.

To describe the time problem of the collision of the bench dragons, this paper establishes a Cartesian coordinate system, and abstracts each handle of the bench dragons as a series of points, in which the front handle of the dragon's head is point A. The jth vertex of the ith bench is defined as P_{ij}. These points are connected sequentially to form a series of vectors, and the unit direction vector and unit normal vector are determined for each vector, which are used to analyze the motion characteristics of the bench dragon as shown in Figure 4 and Figure 5.

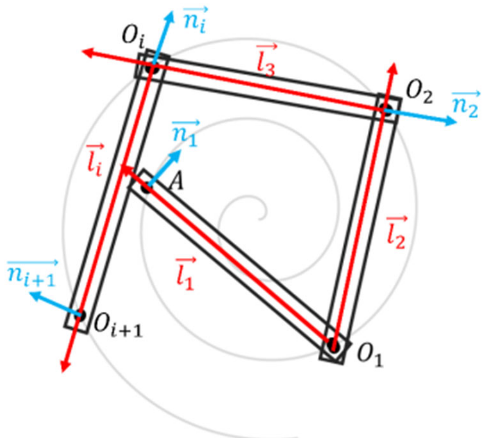

Figure 4. Schematic diagram of the collision of bench dragons

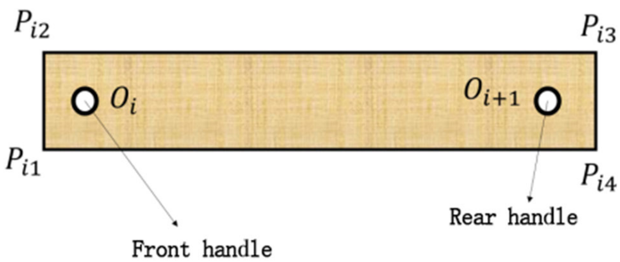

Figure 5. Schematic diagram of bench apexes

Let A (x_0, y_0), O_1 (x_1, y_1), ..., and O_i (x_i, y_i), then the tangential and normal unit vectors of the faucet stool are:

$$\begin{aligned}\vec{e}_1 &= \left(\frac{x_0 - x_1}{|\vec{l}_1|}, \frac{y_0 - y_1}{|\vec{l}_1|}\right)\\ \vec{n}_1 &= \left(\frac{y_0 - y_1}{|\vec{n}_1|}, \frac{x_0 - x_1}{|\vec{n}_1|}\right)\end{aligned} \tag{7}$$

The tangential and normal unit vectors of the dragon body stool and the dragon tail stool are:

$$\begin{aligned}\vec{e}_l &= \left(\frac{x_i - x_{i-1}}{|\vec{l}_i|}, \frac{y_i - y_{i-1}}{|\vec{l}_i|}\right)\\ \vec{n}_l &= \left(\frac{y_i - y_{i-1}}{|\vec{n}_i|}, -\frac{x_i - x_{i-1}}{|\vec{n}_i|}\right)\end{aligned} \tag{8}$$

where $|\vec{l_i}|$ is the distance between the two handles of the bench, the dragon head bench is 286cm, the dragon body bench and the dragon tail bench are 165cm.

3.2. Solving the model

The four vertices of the ith bench P_{ij} (j=1,2,3,4) have coordinates:

$$\begin{aligned}(P_{i1}x, P_{i1}y) &= (O_i x, O_i y) + 0.275\vec{e}_l + 0.15\vec{n}_l,\\ (P_{i2}x, P_{i2}y) &= (O_i x, O_i y) + 0.275\vec{e}_l - 0.15\vec{n}_l,\\ (P_{i3}x, P_{i3}y) &= (O_{i+1}x, O_{i+1}y) - 0.275\vec{e}_{l-1} + 0.15\vec{n}_{l-1},\\ (P_{i4}x, P_{i4}y) &= (O_{i+1}x, O_{i+1}y) - 0.275\vec{e}_{l-1} - 0.15\vec{n}_{l-1}.\end{aligned} \tag{9}$$

The MATLAB inpolygon function can be realized to determine whether the point to be measured is within the specified polygon.

Substitute the four vertices of the bench P_{i1}, P_{i1}, P_{i1}, P_{i1} Substitute the four vertices of the bench into the function to form a polygon, and determine whether the vertices of the dragon's head bench P_{11}, P_{12} are inside the polygon, you can determine whether the dragon head bench collides with the surrounding dragon body benches.

Change the time traversal step to 0.1 seconds, get the positional coordinates of each section of the dragon body and dragon tail every 0.1 seconds, use the above algorithm to calculate the coordinates of the four vertices of each bench around the dragon head, and carry out the collision detection every 0.1s, that is, judge the point to be measured every 0.1s P_{11} and P_{12} whether it is in the quadrilateral P_{i1} P_{i2} P_{i3} P_{i4} (i=2,3,...)'s boundary or interior.

The run revealed a collision at sec 412.5.

At time point 412.5 s, the positions, and velocities of the front handle of the dragon head, the front handles of the 1st, 51st, 101st, 151st, and 201st dragon-body sections after the dragon head, and the rear handle of the dragon's tail are shown in the table. The velocity data show that the velocity of the dragon head is 0.9998 m/s, the velocity of the 1st dragon body is 0.9913 m/s, the velocity of the 51st dragon body is 0.9766 m/s, the velocity of the 101st dragon body is 0.9743 m/s, the velocity of the 151st dragon body is 0.9734 m/s, and the velocity of the 201st dragon body is 0.9729 m/s. The position information shows that the coordinates of the dragon's head are (1.2279, 1.9305), the coordinates of the 1st section of the dragon's body are (-1.6275, 1.7675), the coordinates of the 51st section of the dragon's body are (1.3014, 4.3201), the coordinates of the 101st section of the dragon's body are (-0.5573, -5.8779), the coordinates of the 151st section of the dragon's body are (0.9478, - 6.9601), and the coordinates of dragon body section 201 are (-7.8962, -1.2098).

The geometric relations established on this basis are known:

$$L^2 = r_i^2 + r_{i+1}^2 - 2 \times r_i r_{i+1} \cos(\theta_i - \theta_{i+1}) \tag{10}$$

Substituting$\theta_i = \frac{2\pi}{p} r_i$ Substitute to get:

$$L^2 = r_i^2 + r_{i+1}^2 - 2 \times r_i r_{i+1} \cos\left(\frac{2\pi}{p}(r_i - r_{i+1})\right) \tag{11}$$

This is the recursive formula for the polar diameter of each bench handle, whose initial value r_1 is the front handle pole diameter when the dragon head bench enters the turnaround space, that is, the radius of the turnaround space is 4.5 m. If the dragon head can be smoothly coiled into the turnaround space, the dragon head will not collide with the surrounding dragon body when the front handle of the dragon head is equal to the pole diameter of 4.5 m.

To search for suitable pitch values, the upper bound of pitch values is set to 55cm and the lower bound to 30cm, and all pitch values are traversed within this range in steps of 1mm.

From the recursive formula, it can be seen that each spacing value corresponds to a set of bench dragon's position information, traversing the spacing process, the introduction of the established model can get the bench dragon's position information under different spacings of the Panlong team, and through the introduction of the established model to judge whether the spacing value will collide with the bench dragons, and then to get the smallest spacing point that can enter the headway space without collision, and the smallest spacing is 43.4 cm.

4. Turnaround Curve Optimization

4.1. Modeling

The focus of the study was to optimize the turnaround curve to make the path of the dragon dance team shorter. The turnaround path consists of two tangent arcs, with the radius of the first arc being twice that of the second. The current design requires the curves to be tangent to the disk-in and disk-out solenoids, and it is necessary to analyze whether the turnaround curve can be shortened while maintaining tangency. Through geometric methods and mathematical derivations, the existing curve was adjusted to design a more compact turnaround path. The goal is to improve the efficiency of the turnaround process and shorten the path within the existing constraints. It is also necessary to determine the maximum speed of the front handle of the dragon head when the dragon dance team is traveling, while ensuring that the speed of the other nodes does not exceed 2 m/s. Since the speed of the dragon head affects the speed of the subsequent nodes, a model needs to be built to simulate the movement pattern of the dragon dance team to ensure that the speed of each node conforms to the constraints. This requires an in-depth study of the dynamic behavior of the dragon body, especially when turning the head and coiling out, balancing the speed of the dragon head with the overall coordination. Through analysis, the maximum traveling speed of the dragon's head is derived while meeting the overall safety requirements.

The turnaround path consists of two tangent arcs, and the radius of the first arc is the radius of the second arc. r_1 is twice the radius of the latter arc r_2 is twice the radius of the latter arc, i.e.,$2r_2 = r_1$ The radius of the first arc is twice the radius of the second arc, i.e. These arcs are tangent to the solenoidal path, so the path has good geometric continuity.

A circular arc path can be defined by its center angle θ_1 and θ_2 and the corresponding radii r_1 and r_2 to describe it. According to the circular arc length equation:

$$L = r_1\theta_1 + r_2\theta_2 \tag{12}$$

The total length L of the header path can be expressed as the sum of the lengths of the two arc segments.

Objective optimization equation

Our goal is to minimize the length of the turnaround path L. To do this, we need to determine the radius of each arc r_1 and the angle of the center of the circle θ_1 , and construct the objective function using these as optimization variables:

$$\min f(r_2, \theta_2) = 2r_2\theta_1 + r_2\theta_2 \tag{13}$$

which $\theta_2 = 2\theta_1$, by optimizing r_2 and θ_2 to minimize the path length.

4.2. Solving the model

Before proceeding to the complex optimization, this paper uses traditional optimization methods such as gradient descent and Newton's method to solve the initial optimization problem [6]. The initial parameters of the turnaround path are obtained by deriving the objective function $f\ (r_2, \theta_2)$. To further optimize the length of the turnaround curve, feed-forward neural network (FNN) is introduced for iterative solution. The specific process includes: the input layer receives the initial parameters of the turnaround curve (arc radius r_2, centroid angle θ_2); the hidden layer gradually adjusts the path parameters through the nonlinear transformations of each layer; and the output layer provides the optimized arc radius and angle. The loss function is defined as the total length of the turnaround curve, and the goal is to minimize this length, and the network weights and parameters are adjusted by the back-propagation algorithm to achieve gradual convergence to the optimal solution. After obtaining the optimal head-turning path, the overall trajectory of the bench dragon is described in conjunction with the solenoidal motion equation. The spiral motion is expressed by the polar coordinate equation: $r_\theta = a + b_\theta$, where a and b are the spiral parameters. The geometry of the turnaround curve needs to be tangent to the solenoid, so the solenoid parameters are further adjusted according to the optimization results of the turnaround path. Based on the optimization results of the head-turning curve and the equations of motion of the screw thread, the specific positions and velocities of each section of the body of the bench dragon (including the dragon head, body and tail), especially the key positions (such as the front handles of the body and the rear handles of the tail of the dragon in the 1st, 51st, 101st, 151st, and 201th sections behind the dragon head) are calculated step by step, and the coordinates and velocities of the body and tail of the bench dragon are calculated by numerical methods at each moment.

In the study, the maximum travel speed of the dragon head needs to be determined to ensure that the speed of all sections of the bench dragon body does not exceed 2 m/s. This problem is an optimization problem with constraints, and the optimal solution is found by an optimization algorithm. The objective is to maximize the velocity of the dragon's head vhead so that the velocity of all sections of the dragon's body vn does not exceed 2 m/s. The constraints are that the velocity of each section of the dragon's body $v_n \leq 2$ m/s. The velocity of the dragon's head, the position of each section of the dragon's body, and the curvature of the turnaround path affect the velocity of the dragon's body, and the change of curvature especially in the process of turnaround leads to the fluctuation of the velocity, therefore the maximum velocity needs to be accurately controlled. The maximum speed of the dragon head needs to be precisely controlled. The goal of the model is to find the maximum speed of the dragon head vhead, so that the speed of all sections of the dragon body v_n meets $v_n \leq$ 2m/s. The fitness function $f(v_{head})$ is used to measure whether the speed of the dragon head is in accordance with the speed limit of each section of the dragon body, which is defined as:

$$f(v_{\text{head}}) = \sum_{n=1}^{N} \max\left(0, v_n(v_{\text{head}}) - 2\right) \tag{14}$$

When the velocity of a section of the dragon body exceeds 2 m/s, the value of the fitness function increases, indicating that the solution is not optimal, and the goal is to minimize the value of the function to find the optimal solution. In the model construction, the constraint $v_n \leq$2m/s determines the upper limit of the dragon head speed, and the maximum speed that satisfies this constraint is gradually found by adjusting the dragon head speed.

To solve this optimization problem, Particle Swarm Optimization (PSO) algorithm is used with the following steps:

Particle representation: each particle represents a candidate leading velocity v_{head} .

Particle swarm size: set the number of particles N. Each particle has its own position (indicating the leading speed) and velocity (indicating the search direction).

Initialize velocity and position: randomly initialize the velocity and position of each particle, with the initial faucet velocity in the range of $0 \leq v_{head} \leq 2$ m/s.

For each particle (i.e., each leading velocity candidate), calculate its corresponding fitness value.

Calculate the speed of each section of the dragon's body v_n (v_{head}), if the velocity of a section of the dragon body exceeds 2 m/s, the value of the fitness function increases, indicating that the solution is poor.

Update particle velocity and position.

Particle Velocity Update: The velocity of the particle is updated according to the following formula:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i - x_i(t)) + c_2 \cdot r_2 \cdot (g - x_i(t)) \tag{15}$$

where w is the inertia coefficient of c_1 and c_2 are the acceleration constants, the r_1 and r_2 are the random coefficients, and p_i is the historical best position of the particle, and g is the global best position of the population.

Particle position update: Adjusts its leading velocity candidate solution by updating the particle's velocity:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{16}$$

The particles gradually approach the optimal solution.

After each iteration, it is checked whether the velocity of the dragon body corresponding to the current particle exceeds 2 m/s. If the velocity of a section of the dragon body exceeds the constraint, the fitness value is increased, and the particle is adapted to the best global and individual solution.

The position and velocity of the particles are updated iteratively until the maximum number of iterations is satisfied or the fitness function reaches a predefined optimal value.

Through many iterations, the particle swarm gradually converges to the optimal solution, i.e., the maximum velocity of the faucet.

When the particle swarm optimization converges, output the maximum travel speed of the dragon head v_{head} and ensure that the velocity of all the nodes of the dragon body v_n do not exceed 2 m/s.

5. Conclusion

In this paper, we successfully constructed a motion model based on geometric properties by deeply studying the motion problem in the spiral path, and combined the particle swarm optimisation algorithm and feed-forward neural network to achieve the optimisation of the path and speed. Through the accurate calculation of MATLAB, we not only describe the motion trajectory in detail, but also solve the collision detection problem effectively, and finally determine the minimum pitch and maximum velocity constraints.

It is shown that the use of advanced computational techniques and machine learning algorithms can significantly improve the modelling accuracy and optimisation efficiency of complex motion systems. This study not only provides an effective solution to the motion problem in spiral paths, but also demonstrates the potential of computer science and technology in the application of complex engineering problems.

Future research can be further expanded in the following aspects: firstly, combining more machine learning algorithms to further improve the prediction accuracy and real-time optimisation ability of the model; secondly, applying the research results to real engineering projects, such as robot path planning and automatic driving systems, to verify their practical effects and application value.

In summary, the research in this paper provides new ideas and methods for motion problems in complex paths, which has important theoretical significance and application prospects.

References

[1] Tong Zhenzhu, He Yundong, Guan Jiahao, et al. Research on broadband frequency doubler based on Archimedean helix superlattice structure[J/OL]. Modern Applied Physics,1-10[2024-10-10]. http: // kns.cnki.net/kcms/detail/61.1491.O4.20240913.1458.002.html.

[2] F.F. Tang,J.W. Hu,Y. Ma,et al. Landslide displacement prediction method based on particle swarm optimisation-support vector machine regression model with recursive feature elimination selection[J/OL]. Industrial Building,1-13[2024-10-10]. http: // kns. cnki. net/ kcms/ detail/ 11. 2068. TU.20241008.1326.002.html.

[3] WANG Suqing, ZOU Xuan, GUO Jinbiao,et al. Research on the optimal width and depth model for predicting eye muscle area and backfat thickness in Duroc boars based on feed-forward neural network[J/OL]. Chinese Journal of Animal Husbandry,1-13[2024-10-10]. https: // doi. org/ 10. 19556/j.0258-7033.20231121-07.

[4] Li Weiyang. Complex motion and time-lag velocity feedback control of a class of relative rotation system[D]. Shanghai Institute of Applied Technology,2015.

[5] ZHAO Zhiguang, QU Xiaoling. Application of MATLAB numerical simulation in the teaching of materials engineering fundamentals--The three-dimensional thermal conductivity process of porous materials as an example[J]. Guangdong Chemical Industry,2024,51(17):195-197+231.

[6] YANG Xiaoyong,LONG Qitan. A prediction model of influencing factors of drilling accuracy based on feed-forward neural network[J]. Equipment Manufacturing Technology, 2024, (07): 16-19.