

Mathematical Modeling and Optimization of Bench Dragon Motion Paths

Zijun Shi[#], Yurun Wei[#], Chengjie Zhang[#]

Faculty of Science, Harbin University of Science and Technology, Harbin, 150080, China

[#]These authors contributed equally.

Abstract: "Bench dragon" is a traditional folk culture activity in Zhejiang and Fujian provinces from China, which has the value of viewing and sports. Participants' benches are connected end to end, forming a dragon shape. To optimize the performance, this paper establishes a mathematical model covering collision detection, pitch optimization, and turning path design. The collision detection adopts the separation axis theorem (SAT algorithm) to calculate the normal vector and projection of the bench-frame to judge whether the intersection and ensure the safety in the dynamic environment. The pitch is optimized using the iterative method to make the bench dragon turn around smoothly without collision, and the optimal pitch is 0.43 meters. The turning path design solves the cut point by combining the circle equation, and optimizes the path by plane geometry to obtain the shortest turning path. The model is accurate and efficient, describing positions and velocities per second from -100 to 100 seconds. Future studies could further consider the complicated situations such as deformation of the bench to accommodate the varied scenes. This manuscript provides theoretical support for the design and optimization of the bench-dragon performance and demonstrates the application value of mathematical modeling in traditional culture.

Keywords: Optimization Algorithm, Motion Prediction, Bench Dragon.

1. Introduction

The bench-dragon also known as Panlong, is a traditional folk culture activity originated in Zhejiang and Fujian, China, with profound historical and cultural deposits. The event is usually held during festivals or special occasions, symbolizing unity, strength, and good luck. What is unique is that it is not performed by traditional dragon-shaped props, but by connecting multiple benches end to end to form a winding dragon shape, and then perform. The dragon shape of the performance is usually composed of dozens to hundreds of benches, which is huge and visually stunning.

In addition to the symbolic significance of entertainment and folk culture, the movement mode of the bench-dragon and its unique snake dancing path also have certain academic research value, especially for the fields of dynamics, space movement model and other fields. This cultural activity not only gives people a strong visual impact, but also provides a practical basis for related scientific and model research.

Nowadays, domestic and foreign scholars have studied the construction of optimization models and evaluated various models. Pardalos et al. [1] use simulated annealing method (SA) and hierarchical analysis method to establish related models based on computation-intensive optimization problems and solve NP-hard optimization problems; Mahesh et al. [2] choose ML model and conduct prediction learning based on decision tree model and multi-model to complete the improvement of machine learning algorithm; Jordan [3] et al. use the development of learning algorithms and theories and the availability of online data and low-cost computing to drive data-intensive machine learning methods; Farooq F[4] et al. use the machine intelligence algorithm and a single learner and an integrated learning model for prediction; Abdullah[5] proposes an improved support vector machine (SVM) model. Solve the evaluation and application of SVM model in the field of mechanical learning; Schonlau et al [6] summarized the default data, Combined with the random forest algorithm

(RF), the model is built to solve the credit risk assessment problem for statistical learning. In the machine learning algorithm, the RF random forest model developed from the decision tree shows a strong generalization ability and stability, which is widely used in the classification and regression problem [7]. However, traditional machine learning methods are often limited by the setting of parameters. For example, BP neural network often causes its emergence to trap into the local optimal solution and low prediction accuracy [8]. The proposal of other algorithms brings a feasible solution to the optimization of traditional machine learning models. Particle swarm algorithm (PSO) [9] and ant swarm algorithm (ACO) [10], as the earliest swarm intelligence algorithm, have become the cornerstone of the development of subsequent swarm intelligence algorithm, such as one-dimensional convolutional neural network (1D-CNN) [11], autoregressive model (AR) [12], state space model [13], etc. At the same time of improving these algorithms, the application of intelligent algorithms to optimize the traditional machine learning model has been in-depth studied by scholars at home and abroad: Mohamed Abdel-Basset proposed that the starfinch optimization algorithm NOA [14] simulates the two behaviors of starfinches collecting and storing food in summer and autumn, and searching for the storage location of food in spring and winter. Mohamed Abdel-Basset et al. proposed the spectral optimization algorithm (LSO) [15] in 2022. The spider bee optimization algorithm (SWO) [16] was proposed by Mohamed Abdel-Basset et al. This algorithm models the hunting, nesting and mating behavior of female spider bees, and has the advantages of fast search speed and high solution accuracy. The energy valley optimization algorithm [17] is a novel meta-heuristic algorithm proposed by MahdiAzizi et al. in 2023, inspired by the principle of the physics about stability and different particle decay modes. Scholars at home and abroad have done some research on the intelligent optimization algorithm, and they have also applied it to many

practical fields.

2. Application Method Description

2.1. Geometric models and mathematical functions

In order to better solve the bench dragon problem and enable the application of the results to a wider field. We first translate some abstract data and its concepts from the bench-dragon problem into mathematical problems, and each part of the bench dragon is regarded as a regular rectangular graph according to the specific needs. Then, we transformed the movement trajectory of the bench dragon into the movement of the rectangular center through some physical models, such as the center of mass motion model, Considering the collision situation as the intersection of the border lines of two rectangular blocks. Last, we properly used some mathematical function equations (such as the Archimedean screw) to establish the trajectory equations and motion equations of Bench-dragon, and the iterative equation of speed and position with time is calculated. Through the proper application of mathematical models, this greatly simplifies the calculation process of the abstract thing.

2.2. Optimize the model

However, for some problems in the Bench-dragon model, it is weak to only rely only on mathematical function model. We also need to consider some existing models, such as optimization model, neural network model, etc., among which the optimization model is particularly important in solving the Bench-dragon problem. Next, we provide a detailed introduction of the optimized model.

2.2.1. Interpretation of the optimization algorithm

The optimization algorithm is the method that minimizes the strategy (the loss function) or calculates the optimal solution of the loss function through the iterative method (the optimization algorithm).

2.2.2. The role of the optimization algorithm

Simply short, the function of the optimization algorithm is to adjust the model parameters and then obtain the minimum loss function. At present, optimization algorithms solve the optimal solution of the loss function in an iterative way.

2.2.3. Use of the optimization algorithm

1) Convex optimization problem: If the amount of data is particularly large, then the gradient calculation is very time-consuming, so we will choose to use the iterative method to solve it. Each step of the iteration is small, and it is easy to implement.

2) Non-convex problem: it can only be solved by iterative method, and the value of the objective function in each iteration becomes smaller and the optimal solution is constantly approximated.

In short, the focus of the optimization problem is the iterative method to use, that is to find the iterative formula.

2.2.4. Optimize the process of the algorithm

For a better understanding, we will present the process of optimizing the algorithm in the form of the following flow chart:

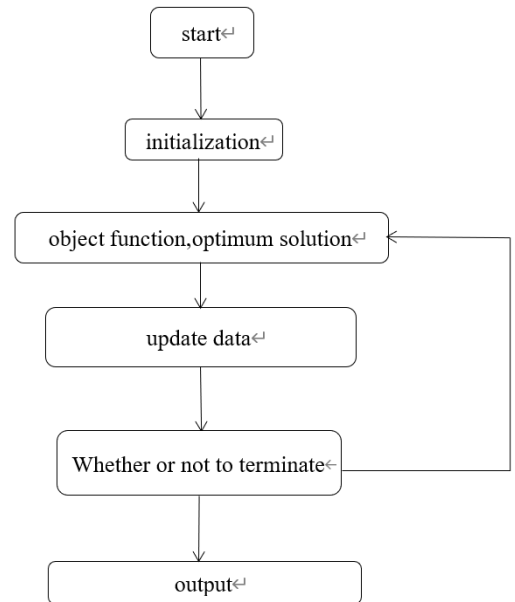


Figure 1. Flow chart of the optimization algorithm

To sum up, the essence of the whole process is to continuously iterate the data until finding the value of the smallest error and meet the requirements.

3. Algorithm Performance Test

3.1. Detection of the bench-dragon collision algorithm

3.1.1. Overview of the SAT algorithm steps

The basic idea of the SAT algorithm is to examine the projections of two convex polygons on a series of separation axes, determine whether the projections overlap, and then infer whether they intersect. The specific steps are described as follows:

The basic idea of the SAT algorithm is to examine the projections of two convex polygons on a series of separation axes, determine whether the projections overlap, and then infer whether they intersect. The specific steps are described as follows:

a. Get the edge normal vector: For each polygon, calculate the normal vector of each edge (the unit vector perpendicular to the edge).

b. Projection operation: project two polygons onto each normal vector to generate one-dimensional line segments (projection results).

c. Check for overlap: Successively check whether the projections of two polygons on all normal vectors overlap. If the projections do not overlap on either separation axis, the two polygons do not intersect in the original space; if all the projections overlap, they intersect.

In conclusion, the SAT algorithm determines whether two convex polygons intersect by finding the separation axes. So long as finding a separation axis such that the projections do not overlap, it is determined that they do not overlap.

3.1.2. Bench dragon collision prediction

First, establish the coordinate equation of each handle of the "bench dragon", and the direction vector of a single bench is calculated according to the coordinates of the front and back handles, thus establishing the direction angle β . Secondly, take the midpoint of the connection between the front and back handles of the bench dragon, which is the two-dimensional geometric center of the bench. Finally, the

position equation of the four endpoints of the bench is found by the function of the midpoint coordinate and the orientation angle β .

3.1.2.1 Establish the bench handle track coordinates and the bench end point track position coordinates

The time step was taken as 1s, calculating the movement trajectory of the "bench dragon" beginning at 301s.

First, indicate the movement direction of the bench, connect the front and rear handles of the bench, and calculate the direction vector of each bench. Thus, the direction angle β is represented according to the direction vector. Then, the relationship between the midpoint coordinate and the direction angle β is calculated through the midpoint coordinate (x_0, y_0) of the front handle and the back handle of the bench.

Taking the movement direction of the head of the dragon as the positive direction of the vector, which is as follows:

Upper left end point of the head of the dragon:

$$\begin{cases} x_a = x_0 - L_0 \cos\beta + d_0 \sin\beta \\ y_a = y_0 - L_0 \sin\beta - d_0 \cos\beta \end{cases} \quad (1)$$

Upper right end point of the head of the dragon:

$$\begin{cases} x_b = x_0 - L_0 \cos\beta - d_0 \sin\beta \\ y_b = y_0 - L_0 \sin\beta + d_0 \cos\beta \end{cases} \quad (1)$$

Lower left end point of the head of the dragon:

$$\begin{cases} x_c = x_0 + L_0 \cos\beta + d_0 \sin\beta \\ y_c = y_0 + L_0 \sin\beta - d_0 \cos\beta \end{cases} \quad (3)$$

Lower right end point of the head of the dragon:

$$\begin{cases} x_d = x_0 + L_0 \cos\beta - d_0 \sin\beta \\ y_d = y_0 + L_0 \sin\beta + d_0 \cos\beta \end{cases} \quad (4)$$

Where $x_{a,b,c,d}$ represent the abscissa of the four vertices of the bench, $y_{a,b,c,d}$ indicates the ordinate of the four vertices of the bench, x_0, y_0 shows the geometrical center of the bench, L_0 express the half the length, d_0 says half the width, and β the orientation angle.

In the process of solving, making use of the relationship between the geometric center and β angle establishes the four endpoint movement models of the bench dragon.

3.1.2.2 Crash detection

Determine the range of benches to check. Because the dragon dance team is spiral, it is speculated that the collision occurred at the position where the dragon head and the dragon body overlapped, and it was detected.

For each bench in the range, use the separation axis to approximate the two benches as line segments and then check whether the two-line segments intersect.

If any collision is detected, stop the time and jump out of the loop. This means that the end of the drive-in moment has been found.

Table 1 details the position and speed of the collision. These data are helpful to analyze the movement trajectory and identify potential collision risk points, thus optimizing the overall performance.

Table 1. The coordinates and velocity of each point of the bench dragon before the collision

	x(m)	y(m)	v(m/s)
DH (fore)	0.785301	2.169331	0.991947
DB (1)	-1.9701	1.406632	0.984423
DB (51)	0.875947	4.43475	0.974993
DB (101)	-0.16801	-5.90762	0.973263
DB (151)	1.285545	-6.91004	0.972628
DB (201)	-7.84967	-1.49901	0.972663
DT	0.707409	8.35019	0.972442

Here, DH and DT are used respectively to replace the dragon head and tail, and DB (i) is used to represent the i-th dragon body.

Figures 1 and 2 show the spatial configurations of bench dragons at seconds 301 and 413. Figure 1 shows the arrangement of bench dragons during normal movement, while Figure 2 shows the specific location where the collision occurred. Through these graphs, we can intuitively understand the movement trajectory and collision situation of bench dragon and provide an intuitive reference for optimizing the model.

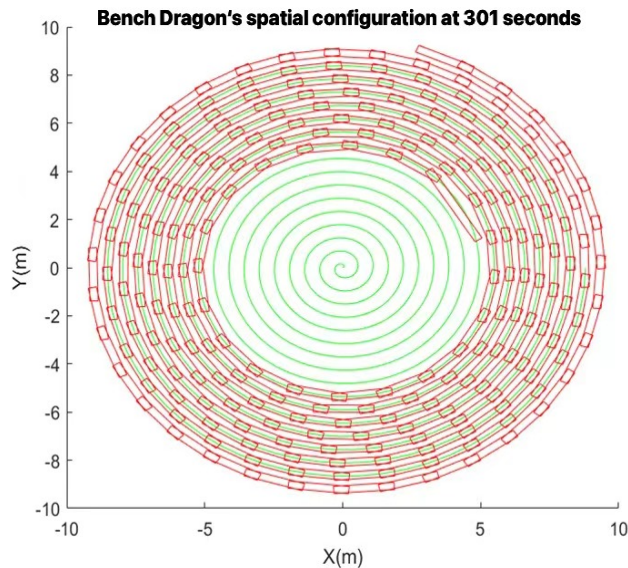


Figure 2. Bench dragon space configuration at 301 seconds

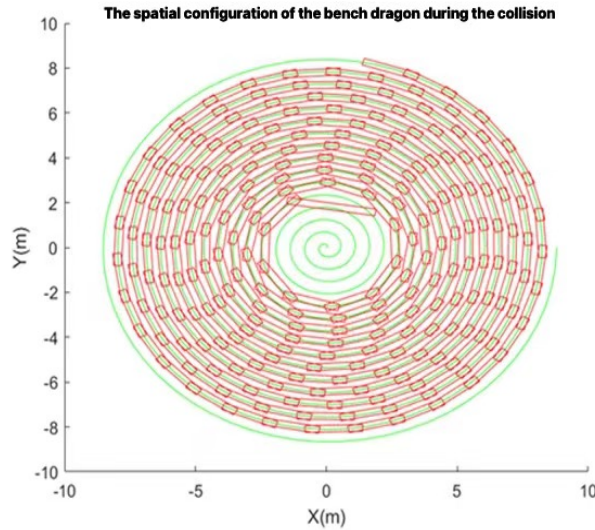


Figure 3. Spatial configuration of the bench dragon at the time of the collision in the 413s

3.2. Screw optimization algorithm testing

Based on the previously established model, this paragraph shows once again:

Spiral model:

$$\mathbf{r}'(\theta) = \mathbf{a} + \mathbf{b}\theta \quad (2)$$

Where \mathbf{b} is the screw-line coefficient, $\mathbf{b} = \frac{\text{pitch}}{2\pi}$, where \mathbf{a} is the pole diameter of the initial position, which $\mathbf{r}(\theta)$ is the pole diameter.

The head of the dragon movement model:

$$\frac{d\theta'}{dt} = -\frac{1}{k\sqrt{1+\theta^2}} \quad (3)$$

Equation for the distance between two points:

$$\sqrt{(x'_{i+1,j} - x'_{i,j})^2 + (y'_{i+1,j} - y'_{i,j})^2} = \begin{cases} L_{i+1,i} = 2.86\text{m} & i = 1, \\ L_{i+1,i} = 1.65\text{m} & i \neq 1, \end{cases} \quad (4)$$

Where i represents the serial number of the "bench dragon" handle, $i = 1, 2, \dots, 223, 224$; j indicates the time node. $x_{i,j}$ indicates the position of the i handle at the j second and the

$L_{i+1,i}$ indicates the distance between the i th handle and the $(i+1)$ th handle.

Collision detection mechanism: Every time the new bench position is calculated, the collision detection must be conducted to ensure that there is no position overlap or contact with the existing bench. Since the dragon dance team uses a spiral arrangement, it is not only necessary to detect the spacing between the new benches and the adjacent benches, but also to pay attention to the relative position with other benches in the outer layer to ensure that there will be no crossover or interference.

Termination condition judgment: Two termination conditions need to be judged when the model runs. First, check that the handle at the front of the tap has reached the boundary of the turn area; second, confirm that there is no collision between the benches. If the tap successfully reaches the boundary and maintains a safe spacing between the benches, the pitch is considered acceptable.

Select the shortest pitch: select the shortest pitch in the qualified pitch set.

The optimization model combines the movement characteristics, structural layout and spatial constraints of the dragon dance team, which can effectively simulate its spiral forward process. When the pitch is 0.42m, the bench dragon will collide, as shown in Figure 4 and Figure 5:

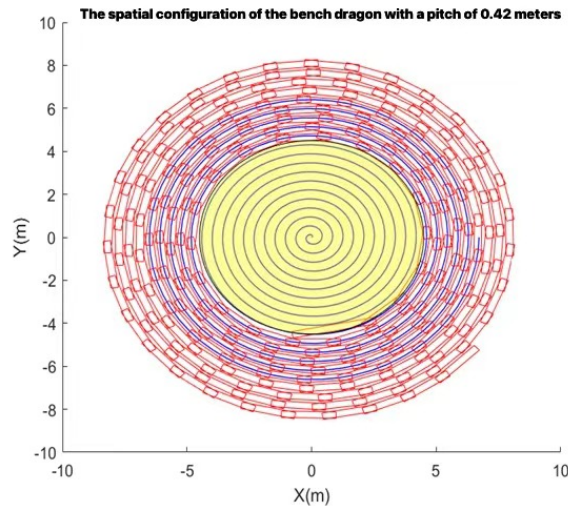


Figure 4. The spatial configuration of the bench dragon at the pitch of 0.42m

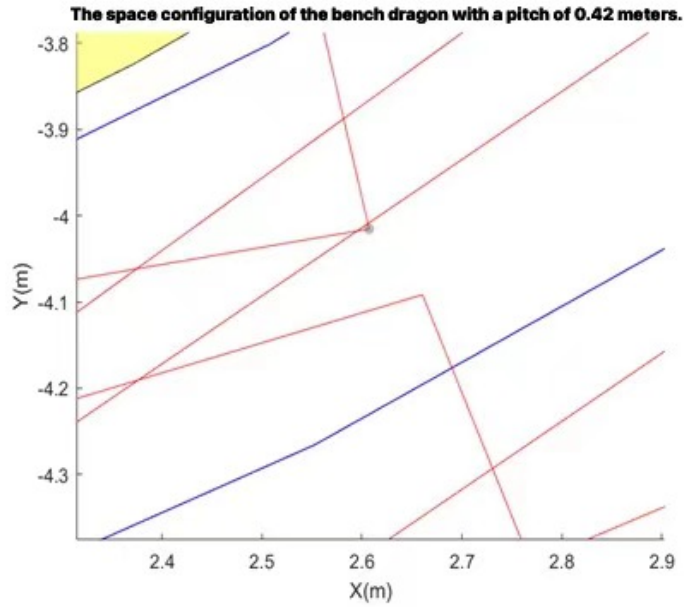


Figure 5. The enlarged version of spatial configuration of the bench dragon at the pitch of 0.42m

Figures 4 and 5 show the bench dragon spatial configuration when the pitch is 0.42 m. Figure 9 shows the overall arrangement, and Figure 10 amplifies the local region where the collision occurs. Through these figures, we can clearly see the specific location of the collision when the pitch is 0.42 meters, which provides intuitive evidence for determining the optimal pitch.

In this study, the pitch was optimized by iterative method to ensure that the bench dragon could enter the U-turn space smoothly without collision. Using the screw model, faucet motion model and collision detection mechanism, the optimal pitch is 0.43 m, which meets the requirements of safe and smooth operation.

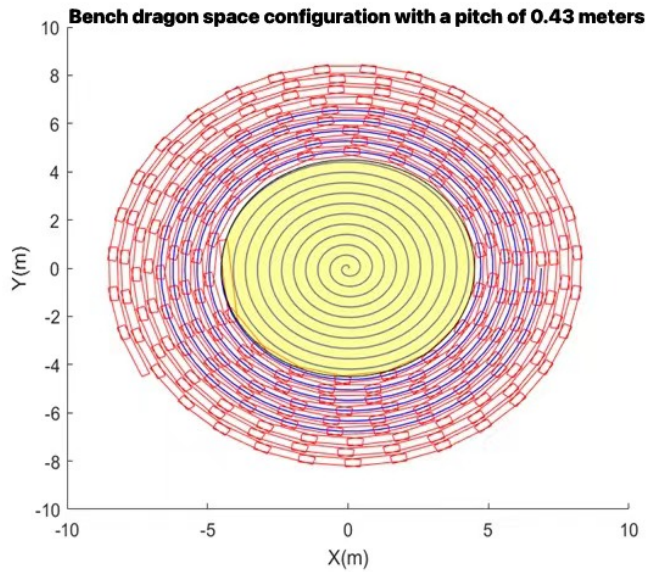


Figure 6. Bench dragon space configuration with a pitch of 0.43 m

Figure 6 shows the bench dragon space configuration with a pitch of 0.43 m. At this time, the bench dragon can smoothly enter the U-turn space without collision, indicating that 0.43 meters is the optimal pitch.

3.3. Planning of the U-turn curve path

Therefore, the model constructed in this paper can optimize and calculate the shortest path while ensuring the tangential movement path when the bench dragon turns around, so the model is reasonably assumed: the pitch of the coiled spiral is 1.7 meters, and the coiled spiral and coiled spiral are centrally symmetrical with respect to the center of the spiral. The dragon dance team completes the U-turn in the U-turn space,

and the U-turn path is composed of two tangent arcs, the radius of the first arc is twice that of the next one, and it is tangent to the coiled in and out spirals.

3.3.1. Turnaround Curve Model Steps Overview

1) Determine the Center Coordinates

Given: Coordinates of point A (x_1, y_1) , coordinates of point B (x_2, y_2) , radius of the smaller arc R, and radius of the larger arc 2R.

The coordinates of the center O_1 of the larger arc can be determined based on the coordinates of point A and the radius of the larger arc 2R. Since O_1 is perpendicular to the tangent at point A and the length of AO_1 is 2R, the coordinates of O_1

can be expressed as:

$$\mathbf{O}_1 = (y_1 + 2R \cdot \sin(\theta_1), y_1 - 2R \cdot \cos(\theta_1)) \quad (5)$$

The coordinates of the center O_2 of the smaller arc can be determined based on the coordinates of point B and the radius of the smaller arc R . Since BO_2 is perpendicular to the tangent at point B and the length of BO_2 is R , the coordinates of O_2 can be expressed as:

$$\mathbf{O}_2 = (x_2 + R \cdot \sin(\theta_2), y_2 - R \cdot \cos(\theta_2)) \quad (6)$$

2) Solve for the Tangent Point C

Let the coordinates of point C be (x_3, y_3) . The equation of the larger circle is:

$$(x - x_{o1})^2 + (y - y_{o1})^2 = (2R)^2 \quad (7)$$

The equation of the smaller circle is

$$(x - x_{o2})^2 + (y - y_{o2})^2 = R^2 \quad (8)$$

By solving these two equations simultaneously, the coordinates of the tangent point C (x_3, y_3) can be obtained.

3) Calculate Angles α and r

The angle α between \mathbf{AO}_1 and \mathbf{CO}_1 can be calculated using the dot product formula of vectors:

$$\alpha = \cos^{-1} \left(\frac{(x_3 - x_1) \cdot (x_{o1} - x_1) + (y_3 - y_1) \cdot (y_{o1} - y_1)}{2R \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}} \right) \quad (9)$$

The angle r between \mathbf{BO}_2 and \mathbf{CO}_2 can be calculated using the dot product formula of vectors:

$$r = \cos^{-1} \left(\frac{(x_3 - x_2) \cdot (x_{o2} - x_2) + (y_3 - y_2) \cdot (y_{o2} - y_2)}{R \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}} \right) \quad (10)$$

4) Solve for the Arc Lengths

The length of the larger arc AC can be calculated using the angle α :

$$\text{arc length}(AC) = 2R \cdot \alpha \quad (11)$$

The length of the smaller arc BC can be calculated using the angle r :

$$\text{arc length}(BC) = R \cdot r \quad (12)$$

5) Calculate the Total Arc Length

$$\text{total arc length} = 2R \cdot \alpha + R \cdot r \quad (13)$$

Assuming that the bench dragon begins to turn as it follows the spiral into the turning area and continues its motion along the spiral out of the turning area as soon as it exits the turning area. By solving the equations of the turning area and the spiral into the area, we obtain the coordinates of point A; by solving the equations of the turning area and the spiral out of the area, we obtain the coordinates of point B. The shortest turning path can be easily calculated as 13.6212 meters.

The region where the bench dragon spirals in, spirals out, and turns around has been plotted using MATLAB, as shown in Figure 7.

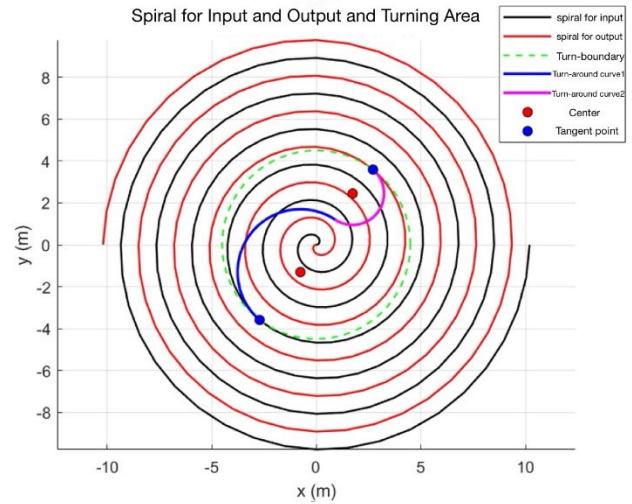


Figure 7. Entrance and exit areas, and turnaround zones

Table 2 records the positions and velocities of each handle of the bench dragon at $t = -100$ seconds and $t = 100$ seconds. These data not only assist in analyzing the motion trajectory of the bench dragon but also provide detailed data support for optimizing its collision detection and pitch.

Table 2. Position and speed of dragon's bench handles at different times

	$t = -100s$	$t = 100s$
DH	(7.7780, 3.7172, 1.0000)	(-3.0835, 7.5761, 1.0000)
DB (1)	(6.2093, 6.1085, 0.9999)	(-0.2681, 8.0795, 1.0001)
DB (51)	(-10.6080, 2.8315, 0.9993)	(2.1626, 3.9926, 1.0040)
DB (101)	(-11.9228, -4.8024, 0.9991)	(-7.3146, 1.9821, 1.0544)
DB (151)	(-14.3510, -1.9810, 0.9989)	(9.4944, -3.4608, 1.0535)
DB (201)	(-11.9529, 10.5670, 0.9988)	(8.4646, 8.6684, 1.0531)
DT	(-1.0111, -16.5276, 0.9988)	(-10.9365, -6.8437, 1.0530)

4. Conclusion

The model successfully integrates the motion of the Bench Dragon with collision detection, accurately determining the time point at which the motion terminates using dynamic

algorithms. This plays a critical role in understanding the motion boundaries of the Bench Dragon and optimizing the performance effect. The model employs efficient algorithms such as spatial partitioning and vectorized computation, significantly enhancing the calculation speed and enabling

large-scale collision detection tasks to be processed within a reasonable time frame, ensuring the rapidity and practicality of the calculations.

The model innovatively formalizes the pitch optimization problem into a mathematical model and combines it with motion simulation, successfully calculating the minimum feasible pitch, which significantly improves the design accuracy of the Bench Dragon performance. The optimal pitch is determined to be 0.43 meters. The model not only considers practical constraints such as collision avoidance and turning space, but also effectively enhances the practicality and reliability of the design through optimization algorithms. This comprehensive and precise design approach provides strong theoretical support for performance planning, allowing designers to maximize performance effects within limited spaces and conditions.

The model cleverly transforms the curve optimization problem into a plane geometry problem, using functional relationships and reasonable assumptions to successfully represent the turning curve. By employing the simulated annealing optimization method, the model solves the shortest turning curve of the Bench Dragon under restrictive conditions, improving the precision of the design. The model can also provide detailed descriptions of the positions and speeds of the Bench Dragon at every second from -100 seconds to 100 seconds, further enhancing the design accuracy. The shape of the Bench Dragon is simplified to line segments or rectangles, ignoring more complex collision scenarios such as the bending and deformation of the benches.

In summary, the model demonstrates strong theoretical support and practical value in the design and optimization of Bench Dragon performances. Especially in areas such as motion and collision detection, pitch optimization, and turning curve design, it provides precise and efficient solutions. However, there is still room for improvement in handling complex collisions and practical operational errors to better adapt to the diverse real-world performance scenarios.

References

- [1] PardalosPM, MavridouTD. Simulated annealing [M] //Encyclopedia of Optimization. Cham: Springer International Publishing, 2024: 1-3.
- [2] Mahesh B. Machine learning algorithms-a review[J]. International Journal of Science and Research (IJSR).[Internet], 2020, 9(1): 381-386.
- [3] Jordan M I, Mitchell T M. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255-260.
- [4] Farooq F, Ahmed W, Akbar A, et al. Predictive modeling for sustainable high-performance concrete from industrial wastes: A comparison and optimization of models using ensemble learners[J]. Journal of Cleaner Production, 2021, 292: 126032.
- [5] Abdullah D M, Abdulazeez A M. Machine learning applications based on SVM classification a review[J]. Qubahan Academic Journal, 2021, 1(2): 81-90.
- [6] Schonlau M, Zou R Y. The random forest algorithm for statistical learning[J]. The Stata Journal, 2020, 20(1): 3-29.
- [7] Aghware F O, Ojugo A A, Adigwe W, et al. Enhancing the random forest model via synthetic minority oversampling technique for credit-card fraud detection[J]. Journal of Computing Theories and Applications, 2024, 1(4): 407-420.
- [8] Qin X, Liu Z, Liu Y, et al. User OCEAN personality model construction method using a BP neural network[J]. Electronics, 2022, 11(19): 3022.
- [9] Gad A G. Particle swarm optimization algorithm and its applications: a systematic review[J]. Archives of computational methods in engineering, 2022, 29(5): 2531-2561.
- [10] Tan Y, Ouyang J, Zhang Z, et al. Path planning for spot welding robots based on improved ant colony algorithm[J]. Robotica, 2023, 41(3): 926-938.
- [11] Harbola S, Coors V. One dimensional convolutional neural network architectures for wind prediction[J]. Energy Conversion and Management, 2019, 195: 70-75.
- [12] Yu J, Xu Y, Koh J Y, et al. Scaling autoregressive models for content-rich text-to-image generation[J]. arXiv preprint arXiv:2206.10789, 2022, 2(3): 5.
- [13] Li K, Li X, Wang Y, et al. Videomamba: State space model for efficient video understanding[C]//European Conference on Computer Vision. Springer, Cham, 2025: 237-255.
- [14] Abdel-Basset M, Mohamed R, Jameel M, et al. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems[J]. Knowledge-Based Systems, 2023, 262: 110248.
- [15] Abdel-Basset M, Mohamed R, Sallam KM, Chakraborty RK. Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm. Mathematics. 2022, 10(19): 3466.
- [16] Abdel-Basset M, Mohamed R, Jameel M, et al. Spider wasp optimizer: a novel meta-heuristic optimization algorithm[J]. Artificial Intelligence Review, 2023, 56(10): 11675-11738.
- [17] Azizi M, Aickelin U, A. Khorshidi H, et al. Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization[J]. Scientific Reports, 2023, 13(1): 226.