

Research Article

Analysis and Optimization of Interpolation Points for Quadruped Robots Joint Trajectory

Mingfang Chen , **Kaixiang Zhang** , **Sen Wang** , **Fei Liu**, **Jinxin Liu**,
and **Yongxia Zhang** 

Faculty of Mechanical and Electrical Engineering, Kunming University of Science and Technology, Kunming, Yunnan, China

Correspondence should be addressed to Sen Wang; wangsen0401@126.com

Received 27 April 2020; Accepted 25 May 2020; Published 1 July 2020

Guest Editor: Qiang Chen

Copyright © 2020 Mingfang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Trajectory planning is the foundation of locomotion control for quadruped robots. This paper proposes a bionic foot-end trajectory which can adapt to many kinds of terrains and gaits based on the idea of trajectory planning combining Cartesian space with joint space. Trajectory points are picked for inverse kinematics solution, and then quintic polynomials are used to plan joint space trajectories. In order to ensure that the foot-end trajectory generated by the joint trajectory planning is closer to the original Cartesian trajectory, the distributions of the interpolation point are analyzed from the spatial domain to temporal domain. An evaluation function was established to assess the closeness degree between the actual trajectory and the original curve. Subsequently, the particle swarm optimization (PSO) algorithm and genetic algorithm (GA) for the points selection are used to obtain a more precise trajectory. Simulation and physical prototype experiments were included to support the correctness and effectiveness of the algorithms and the conclusions.

1. Introduction

Trajectory planning is the first step of quadruped robots towards complex environments [1, 2]. At present, the trajectory planning is mainly divided into two categories: Cartesian space trajectory planning and joint space trajectory planning [3, 4]. The former focuses on the macro area, while the latter pays attention to the micro area. The Cartesian space trajectory planning involves the actual motion trajectory of actuator in three-dimensional space, while the joint space trajectory planning studies the connection problem of joint angles in the process of motion [5]. Hence, for foot-end trajectory planning of quadruped robots, combining the above two methods can lay a superior foundation for its flexible movement in real time.

For the purpose of making the foot-end perform more flexible in Cartesian space, Sakakibara et al. proposed a compound cycloid method and planned a smooth foot trajectory for their walking robot [6]. Based on their method, Wang et al. derived a zero-impact trajectory and the cycle control strategy for one-leg gait [7]. On the other hand, Chevallereau et al. proposed a ballistic trajectory [8], which followed the principle of alternation of muscle movements. Vundavilli et al.

constructed a foot-end trajectory based on a cubic polynomial and planned an optimal gait for dynamic balance of biped robots [9]. In order to make the foot-end trajectory more biological, Kim et al. proposed an elliptical trajectory to simulate the backswing and retraction characteristics of animals when they departed or touched the ground [10]. Semini et al. added the foot speed in the flattened part of the semi-ellipse to coordinate the forward speed of the trunk during the trot motion [11]. Rong et al. used a compound trajectory consisting of cubic curves with straight lines on their Scalf robot [12] and demonstrated a foot-end trajectory consisting of five-degree polynomials with four-degree polynomials in Scalf afterward [13]. For more flexibility, the researchers of MIT used the Bézier curve in its MIT Cheetah [14], which utilized twelve position control points for the foot-end trajectory and connected them with eleven-degree Bernstein polynomial curve. The works of these scholars have provided rich theoretical foundations for the research of the foot-end trajectory of the legged robots and have developed the trajectory in the direction of high smoothness, high bionics, and strong adaptability. However, the trajectory planning in Cartesian space requires repeated inverse kinematics calculations during movement. As the complexity of the foot-end trajectory increases, it will cost a

large computational load on the controller. Consequently, in the process of movement, if the accuracy of the trajectory is guaranteed, the numbers of trajectory transition points will increase subsequently, but the real-time performance of the locomotion will be reduced. Therefore, joint space trajectory planning can alleviate the contradiction between motion accuracy and real-time in the Cartesian space trajectory planning to some extent.

Joint space trajectory planning is a way for locomotion via sampling the designed trajectory and extracting few trajectory intermediate points to execute inverse kinematics solution. And then some smooth curves are used to connect the intermediate points of the joint angles inversely solved by kinematics [15]. For getting the smooth connections between the intermediate points, Liu et al. proposed a method for generating joint trajectories based on X-splines and quartic polynomials [16]. In the Robotics Toolbox, Corke utilized a fifth-degree polynomial to perform point-to-point joint space trajectory planning and obtained a smooth curve between two points [17]. Dehghani et al. also used this method to control the action of the five-link biped, ensuring continuous angular velocity and angular acceleration during locomotion [18]. In order to make the robot's locomotion process more stable, Gasparetto et al. endeavored to optimize the joint motion time and jerkiness to ensure that the trajectory has sufficient smoothness [19]. Liu et al. used a combination of Cartesian space multi-order spline function and joint space multi-order B-spline function to ensure continuous jerk while improving the smoothness of trajectory tracking control [20]. Zhong et al. performed 8-degree polynomial trajectory planning in joint space on the basis of 8-degree polynomial trajectory planning for the center of mass to form a corresponding relationship [21]. In order to avoid the dynamic singularity during the movement, Wang et al. parameterized the joint trajectory into a fifth-order Bézier curve and combined the particle swarm optimization (PSO) to optimize the curve control points [22]. The researches of the above scholars laid the foundation for joint space trajectory planning, which made it perform better in practical applications. Accordingly, combining the trajectory planning methods of Cartesian space with joint space may lead to better trajectory planning actual effects. However, regarding how to coordinate the Cartesian trajectory and the joint trajectory, especially with regard to the choice of middle points, there is less research work done. Secondly, the errors between the actual foot-end curve generated by the joint trajectory motion and the designed Cartesian trajectory need to be further reduced.

In this paper, a bionic foot-end trajectory was designed for the physical quadruped robot platform named MQ robot as shown in Figure 1. The trajectory imitates the actual animal movement laws and divides the foot-end trajectory into three segments: backswing, advance, and retraction. In order to meet the real-time requirements in actual use, joint space trajectory planning was further performed. Eleven

points were picked up from the Cartesian trajectory for inverse kinematics solution, and fifth-degree polynomials were used for piecewise interpolation. In order to explore the effect of different points on the generated actual foot-end trajectory, the intermediate points were taken at equal intervals distribution and equal arc length distributions for comparing and analyzing. Furthermore, according to the characteristics of parameter function, the isochronous points and Chebyshev nodes were taken into the modes of distributions in the temporal domain [23]. Based on the analysis results of the above modes, the PSO algorithm and genetic algorithm (GA) are adopted to generate a better distribution of trajectory points, which use the errors between the actual trajectory and the design trajectory as the fitness values by establishing the models of both trajectories. Further, simulation and prototype experiments were carried on to verify the validity of the analysis and optimization results. The innovations of this paper are as follows: (i) the foot trajectory of the quadruped robots was divided into three segments based on biological observation. (ii) Cartesian space trajectory planning and joint space trajectory planning were combined in order to obtain a better practical application effect. (iii) The distribution of the interpolation points was analyzed and optimized to make the actual trajectory closer to the original design trajectory.

2. Bionic Trajectory Planning of Quadruped Robots

2.1. Cartesian Space Trajectory Planning. In the process of animal movements, the backswing and retraction of the feet are universal, which could ensure the continuity of its movement process [24]. As shown in Figure 2, according to the characteristics during the actual movement of animals, this paper utilizes a bionic foot-end trajectory for the quadruped robots. Through the trajectory decomposition, the foot-end trajectory of Cartesian space is decomposed into parameter functions of the forward direction X and the lifting direction Y with the parameter t (time), and then piecewise quintic polynomial curves are used for interpolation. Among them, the piecewise curves in the X direction have correspondingly added backswing and retraction segments based on the forward segments proposed in [6, 7], and the piecewise curve in the Y direction maintains the characteristics of symmetry. As deduced from quintic polynomials, the trajectory can be gotten by equations (1) and (2). Taking advantage of the quintic polynomial curve which needs six parameters to complete itself, the trajectory should give six initial elements to deduce its form, so that the endpoint position, speed, and acceleration of each curve are independently adjustable to meet the requirements of different terrains and ensure the locomotion process smoothness without impact theoretically:

$$\left\{ \begin{aligned} & \frac{[12S_{S1} - 6v_{s0} \cdot T_{S1} - (a_{s0} - a_{s1}) \cdot T_{S1}^2]}{2T_{S1}^5} \cdot (t - t_0)^5 + \frac{[-30S_{S1} + 16v_{s0} \cdot T_{S1} + (3a_{s0} - 2a_{s1}) \cdot T_{S1}^2]}{2T_{S1}^4} \cdot (t - t_0)^4 \\ & + \frac{[20S_{S1} - 12v_{s0} \cdot T_{S1} - (3a_{s0} - a_{s1}) \cdot T_{S1}^2]}{2T_{S1}^3} \cdot (t - t_0)^3 + \frac{1}{2}a_{s0} \cdot (t - t_0)^2 + v_{s0} \cdot (t - t_0) + s_0, \quad t \in [t_0, t_1] \\ & \frac{[12S_{S2} - (a_{s1} - a_{s2}) \cdot T_{S2}^2]}{2T_{S2}^5} \cdot (t - t_1)^5 + \frac{[-30S_{S2} + (3a_{s1} - 2a_{s2}) \cdot T_{S2}^2]}{2T_{S2}^4} \cdot (t - t_1)^4 \\ & + \frac{[20S_{S2} - (3a_{s1} - a_{s2}) \cdot T_{S2}^2]}{2T_{S2}^3} \cdot (t - t_1)^3 + \frac{1}{2}a_{s1} \cdot (t - t_1)^2 + s_1, \quad t \in [t_1, t_2] \\ & \frac{[12S_{S3} - 6v_{sd} \cdot T_{S3} - (a_{s2} - a_{sd}) \cdot T_{S3}^2]}{2T_{S3}^5} \cdot (t - t_2)^5 + \frac{[-30S_{S3} + 14v_{sd} \cdot T_{S3} + (3a_{s2} - 2a_{sd}) \cdot T_{S3}^2]}{2T_{S3}^4} \cdot (t - t_2)^4 \\ & + \frac{[20S_{S3} - 8v_{sd} \cdot T_{S3} - (3a_{s2} - a_{sd}) \cdot T_{S3}^2]}{2T_{S3}^3} \cdot (t - t_2)^3 + \frac{1}{2}a_{s2} \cdot (t - t_2)^2 + s_2, \quad t \in [t_2, t_d]. \end{aligned} \right. \quad (1)$$

As shown in equation (1), the definitions of the indirect parameters S_{S1} , T_{S1} , S_{S2} , T_{S2} , S_{S3} , T_{S3} (obtained by calculation of direct parameters) are as follows:

$$S_{S1} = s_1 - s_0,$$

$$T_{S1} = t_1 - t_0,$$

$$S_{S2} = s_2 - s_1,$$

$$T_{S2} = t_2 - t_0,$$

$$S_{S3} = s_d - s_2,$$

$$T_{S3} = t_d - t_2,$$

$$y = \left\{ \begin{aligned} & \frac{[12H_{H1} - 6v_{h0} \cdot T_{H1} - (a_{h0} - a_{hw}) \cdot T_{H1}^2]}{2T_{H1}^5} \cdot (t - t_0)^5 + \frac{[-30H_{H1} + 16v_{h0} \cdot T_{H1} + (3a_{h0} - 2a_{hw}) \cdot T_{H1}^2]}{2T_{H1}^4} \cdot (t - t_0)^4 \\ & + \frac{[20H_{H1} - 12v_{h0} \cdot T_{H1} - (3a_{h0} - a_{hw}) \cdot T_{H1}^2]}{2T_{H1}^3} \cdot (t - t_0)^3 + \frac{1}{2}a_{h0} \cdot (t - t_0)^2 + v_{h0} \cdot (t - t_0) + h_0, \quad t \in [t_0, t_w] \\ & \frac{[12H_{H2} - 6v_{hd} \cdot T_{H2} - (a_{hw} - a_{hd}) \cdot T_{H2}^2]}{2T_{H2}^5} \cdot (t - t_w)^5 + \frac{[-30H_{H2} + 14v_{hd} \cdot T_{H2} + (3a_{hw} - 2a_{hd}) \cdot T_{H2}^2]}{2T_{H2}^4} \cdot (t - t_w)^4 \\ & + \frac{[20H_{H2} - 8v_{hd} \cdot T_{H2} - (3a_{hw} - a_{hd}) \cdot T_{H2}^2]}{2T_{H2}^3} \cdot (t - t_w)^3 + \frac{1}{2}a_{hw} \cdot (t - t_w)^2 + h_w, \quad t \in [t_w, t_d], \end{aligned} \right. \quad (2)$$

where the indirect parameters H_{H1} , T_{H1} , H_{H2} , T_{H2} , are defined as follows:

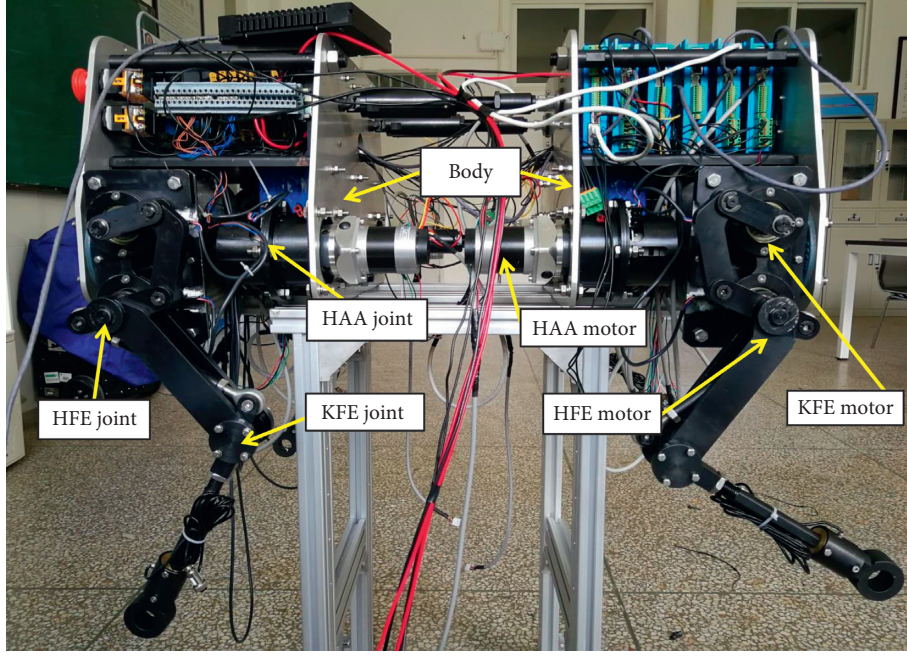


FIGURE 1: Experimental prototype of MQ quadruped robot.

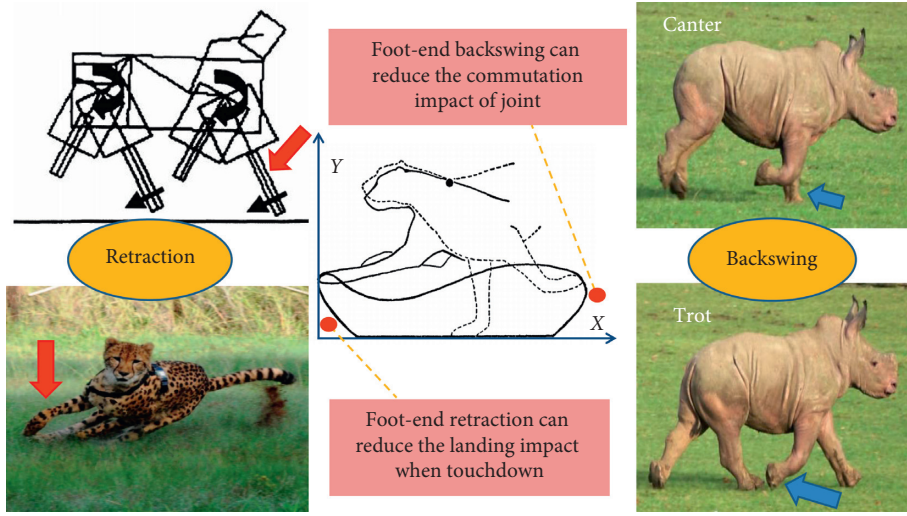


FIGURE 2: The backswing and retraction of feet during animal movements [24–27].

$$\begin{aligned}
 H_{H1} &= h_w - h_0, \\
 T_{H1} &= t_w - t_0, \\
 H_{H2} &= h_d - h_w, \\
 T_{H2} &= t_d - t_w.
 \end{aligned} \tag{3}$$

For the direct parameters (boundary conditions) in equations (1) and (2), the sizes of time t_i and positions s_i , h_i are adjustable according to the environmental terrains, and the speeds v_{si} , v_{hi} and accelerations a_{si} , a_{hi} are adjusted according to the robot's forward speed. In this paper, for

studying the joint space trajectory planning of the foot-end trajectory when crawling on the ground, the values of the above direct parameters are set as shown in Table 1.

In Table 1, the positions can be adjusted according to the needs of terrains and step length. s_0 and h_0 are the starting points of the motion, which coincide with the origin of the coordinate system by default. In actual use, translation can be performed according to the difference in the set coordinate system. The foot-end trajectory after translation is shown in Figure 3. The speed values at t_1 , t_2 , and t_w are 0 by default, so they do not appear in equations (1) and (2). The speed values at t_0 and t_d refer to the speed in [7] for zero

TABLE 1: Boundary conditions of the bionic trajectory.

Parameters	Forward direction X				Lifting direction Y		
Time (s)	$t_0=0$	$t_1=0.25$	$t_2=0.75$	$t_d=1$	$t_0=0$	$t_w=0.5$	$t_d=1$
Position (mm)	$s_0=0$	$s_1=-15$	$s_2=85$	$s_d=70$	$h_0=0$	$h_w=61.8$	$h_d=0$
Velocity (mm/s)	$v_{s0}=0$	0	0	$v_{sd}=0$	$v_{h0}=0$	0	$v_{hd}=0$
Acceleration (mm/s ²)	$a_{s0}=0$	$a_{s1}=1760$	$a_{s2}=-1760$	$a_{sd}=0$	$a_{h0}=0$	$a_{hw}=-1648$	$a_{hd}=0$

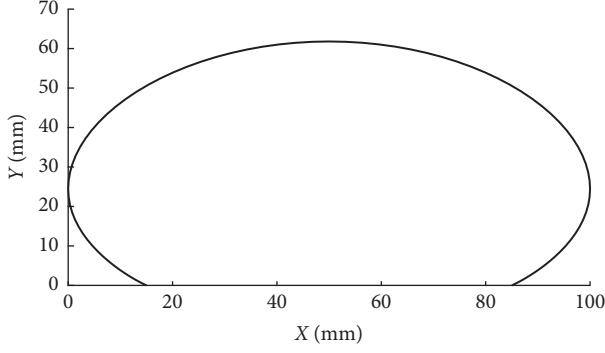


FIGURE 3: Foot-end bionic trajectory in Cartesian space.

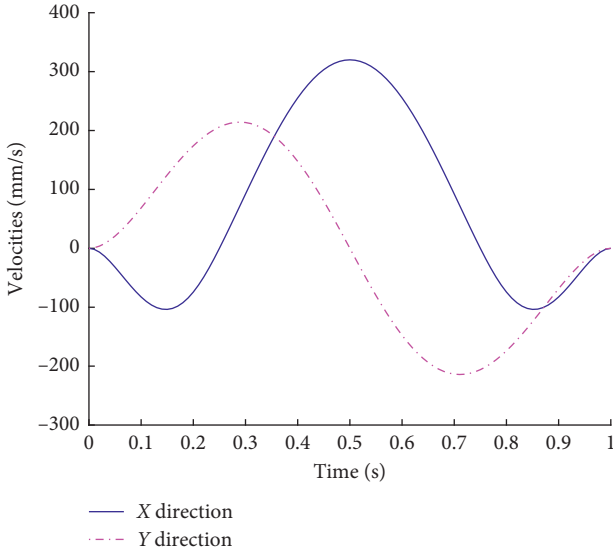


FIGURE 4: Velocity curve of submovement of trajectory.

impact, and the velocity curves of the submotion in two directions are shown in Figure 4. The values of a_{s1} , a_{s2} , and a_{sw} are optimized to obtain smooth acceleration curves as shown in Figure 5.

In Figure 3, the starting point in the X direction is shifted to the right by 15(mm) to facilitate the representation of the foot-end trajectory. In the same way, different translation can be performed on the starting points for different coordinate systems.

2.2. Kinematics Analysis. In order to carry on the joint space trajectory planning in the next discussion, the single-leg structure of the experimental prototype needs to be simplified,

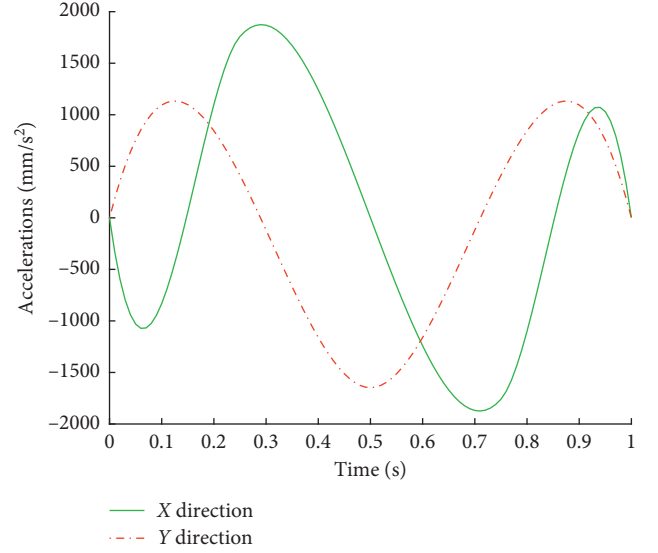


FIGURE 5: Acceleration curve of submovement of trajectory.

and then the kinematics analysis can be performed. As shown in Figure 6, the single-leg structure of the prototype is extracted and compared with the general platform. It is worth mentioning that many researchers had studied the kinematics solution of the general platform [28–30]. On these bases, the relationship between the single-leg drive angle of the prototype and the general platform is shown in the following: where $\hat{\theta}_1$, $\hat{\theta}_2$, and $\hat{\theta}_3$ are the driving joint angles of the actual prototype. Respectively, θ_1 , θ_2 , and θ_3 are the joint angles of the general platform. The other parameters involved are the dimensions of the legs of quadruped robot, as shown in Table 2.

$$\hat{\theta}_1 = \theta_1, \quad (4)$$

$$\hat{\theta}_2 = \theta_2, \quad (5)$$

$$\begin{aligned} \hat{\theta}_3 = & -180 + \theta_2 \\ & + \arccos \frac{2a_4^2 - 2a_4 \cdot l_2 \cdot \cos(\theta_3 - 10)}{2a_4 \cdot \sqrt{a_4^2 + l_2^2 - 2a_4 \cdot l_2 \cdot \cos(\theta_3 - 10)}} \\ & + \arccos \frac{2l_2^2 - 2a_4 \cdot l_2 \cdot \cos(\theta_3 - 10)}{2l_2 \cdot \sqrt{a_4^2 + l_2^2 - 2a_4 \cdot l_2 \cdot \cos(\theta_3 - 10)}}, \end{aligned} \quad (6)$$

2.3. Joint Space Trajectory Planning. Due to the limited computation of the controller [31], in order to alleviate the contradiction between the accuracy of foot-end trajectory and

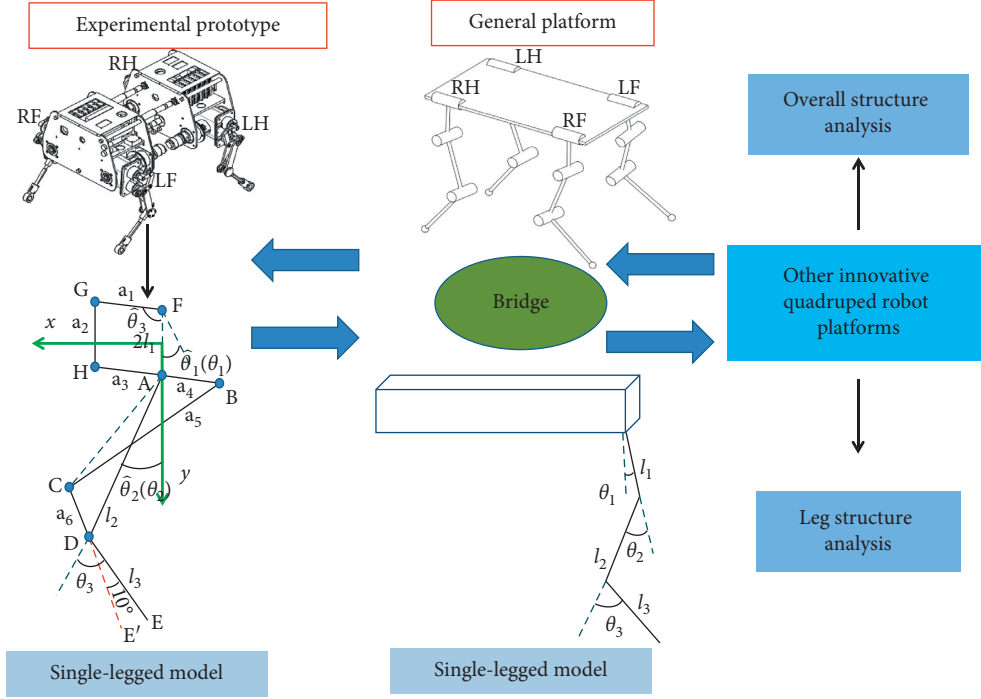


FIGURE 6: Simplification and transformation between the single-leg structure of the prototype and the general platform.

TABLE 2: Properties of the prototype leg.

Parameter	Description	Value (mm)
a_1	Length of a_1	50
$a_2 (2l_1)$	Length of a_2 or $2l_1$	84
a_3	Length of a_3	50
$a_4 (a_6)$	Length of a_4 or a_6	34
$a_5 (l_2)$	Length of a_5 or l_2	170
l_3	Length of l_3	245

the real-time of locomotion, the joint space trajectory planning is carried out for the foot-end trajectory. The movement process of the entire trajectory is divided into a small number of n segments which leads to $n + 1$ points being selected as interpolation points. Therefore, it can reduce the computational burden of the controller to a large extent.

The selections of different interpolation function have an essential influence on the effect of joint space trajectory planning. A good interpolation function can reduce the amount of calculations and improve the accuracy of the movement. At present, there are two kinds of interpolation methods commonly used: high-order interpolation and piecewise low-order interpolation. The high-order interpolation has a large amount of computation, and it is easy to produce the Runge phenomenon which will lead to the lack of accuracy in the beginning and end stages of motion [32]. Segmented low-order interpolation can avoid the occurrence of Runge phenomenon, but due to the order limit, the motion characteristics of each interpolation point cannot be adjusted independently. Therefore, counterpoising the advantages and disadvantages of the two methods, the joint space trajectory planning is carried out by the method of piecewise quintic

polynomial interpolation. It can ensure that the angle, angular velocity, and angular acceleration of each interpolation point are independently controllable, and can correspond with the trajectory planning of Cartesian space in Section 2.1. So it is beneficial to further improving the trajectory accuracy.

Since the motion characteristics of quintic polynomials at each interpolation point can be adjusted independently, the boundary conditions of function curve at both endpoints are set as follows:

$$\begin{aligned}
 \theta(t_0) &= \theta_0, \\
 \dot{\theta}(t_0) &= \omega_0, \\
 \ddot{\theta}(t_0) &= \alpha_0, \\
 \theta(t_f) &= \theta_f, \\
 \dot{\theta}(t_f) &= \omega_f, \\
 \ddot{\theta}(t_f) &= \alpha_f.
 \end{aligned} \tag{7}$$

Taking time t as the independent variable, the function expression of angle $\theta(t)$ is illustrated as

$$\begin{aligned}
\theta(t) = & \frac{(12 \cdot \Delta\theta - (\omega_0 + \omega_f) \cdot 6 \cdot \Delta t - (\alpha_0 - \alpha_f) \cdot \Delta t^2)}{2 \cdot \Delta t^5} (t - t_0)^5 \\
& + \frac{(-30 \cdot \Delta\theta + (16\omega_0 + 14\omega_f) \cdot \Delta t + (3\alpha_0 - 2\alpha_f) \cdot \Delta t^2)}{2 \cdot \Delta t^4} (t - t_0)^4 \\
& + \frac{(20 \cdot \Delta\theta - (12\omega_0 + 8\omega_f) \cdot \Delta t - (3\alpha_0 - \alpha_f) \cdot \Delta t^2)}{2 \cdot \Delta t^3} (t - t_0)^3 \\
& + \frac{\alpha_0}{2} (t - t_0)^2 + \omega_0 (t - t_0) + \theta_0, \quad t \in [t_0, t_f],
\end{aligned} \tag{8}$$

where the indirect parameters $\Delta\theta$, Δt in (8) are defined as follows:

$$\begin{aligned}
\Delta\theta &= \theta_f - \theta_0, \\
\Delta t &= t_f - t_0.
\end{aligned} \tag{9}$$

It can be seen from equation (8) that the interpolation function needs to put in the boundary conditions θ_0 , θ_f , ω_0 , ω_f , α_0 , α_f and then to solve the six unknown parameters in the $\theta(t)$ expression according to these six known parameters. Accordingly, the unique solution of the equation is obtained. Therefore, the reasonable setting of the boundary conditions θ , ω , α is the primary to solve this case. In order to ensure that the function curve is smooth without fluctuation, the boundary conditions are solved using the method of difference quotient and then set as their average value. After calculating the difference quotient of the known angular values θ_{k-1} , θ_k , and θ_{k+1} in order, the average value of the adjacent is taken as the angular velocity ω_k , which is expressed as

$$\omega_k = \frac{(\theta_k - \theta_{k-1})/\Delta t_{k-1} + (\theta_{k+1} - \theta_k)/\Delta t_k}{2}, \tag{10}$$

where Δt_k is the time interval between the time t_k corresponding to the interpolation point k and the time t_{k+1} corresponding to the next interpolation point $k+1$, and Δt_{k-1} is the same definition as Δt_k . Similarly, the solution process of the angular acceleration α_k is shown in the following equation:

$$\alpha_k = \frac{(\omega_k - \omega_{k-1})/\Delta t_{k-1} + (\omega_{k+1} - \omega_k)/\Delta t_k}{2}. \tag{11}$$

For the start and end points of the whole trajectory, ω and α , the boundary parameters, can be specified according to the motion state and forward speed. For the crawling gait, the start and end boundary conditions of ω and α are set as $\omega_0 = 0$, $\omega_n = 0$, $\alpha_0 = 0$, $\alpha_n = 0$. According to the above derivation method, the solution process of boundary conditions at each interpolation point is shown in Figure 7.

3. Selection and Analysis of Interpolation Points

Different interpolation functions can result in different trajectory planning effects; accordingly, selecting different interpolation points can also lead to different trajectory accuracy [33]. However, there are few discussions with the effects caused by different interpolation points distributions.

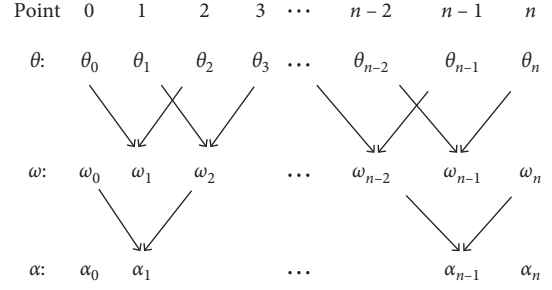


FIGURE 7: Solution process of interpolation point boundary condition.

In order to reduce the number of intermediate points in joint space trajectory planning and have better trajectory reducibility, the selection of interpolation points is particularly important. Appropriate interpolation points can reduce the number of points and increase the proximity to the original trajectory [34]. Based on the above bionic trajectory, the whole trajectory is divided into 10 segments from the spatial domain and the temporal domain, so that the trajectory is symmetrical and the principle of fewer interpolation points is also guaranteed. Accordingly, 11 corresponding trajectory points are obtained. Then, the inverse kinematics solution is used to obtain the interpolation points of joint trajectory.

3.1. Trajectory Division in Spatial Domain. According to the designed Cartesian trajectory, the equidistant and equal arc length are as the method to partition trajectory, and then the corresponding joint trajectory interpolation points are solved by inverse kinematics for analysis. Firstly, the forward direction X of the trajectory is divided into 10 equal parts, and 11 corresponding trajectory points are obtained by turns as shown in Figure 8(a), and then the corresponding joint trajectory interpolation points are obtained by kinematic inverse solution. In the same way, the trajectory points are obtained after dividing the trajectory by equal arc length, as shown in Figure 8(b), and then the inverse kinematics solution is carried out for them to acquire the interpolation points.

In Figure 8, the coordinates of 11 trajectory points obtained by uniform X are shown in Table 3, and the coordinates of 11 trajectory points obtained by uniform arc are shown in Table 4. In Figure 8(a), between points 1, 2, and 3, since X decreases first and then increases, there is intersection between two adjacent intervals. Similarly, there is also intersection between points 9, 10, and 11. Hence, the beginning and end of the trajectory do not appear to be equidistant. However, the change trend and length of equidistant can be obtained from Table 3.

Put the trajectory points in Tables 3 and 4 into equations (4)–(6) for inverse kinematics solution, getting the corresponding joint trajectory interpolation points, and then put the obtained interpolation points into equations (10) and (11), getting the boundary angular velocity and acceleration of each point. Then put them into equation (8) for the joint

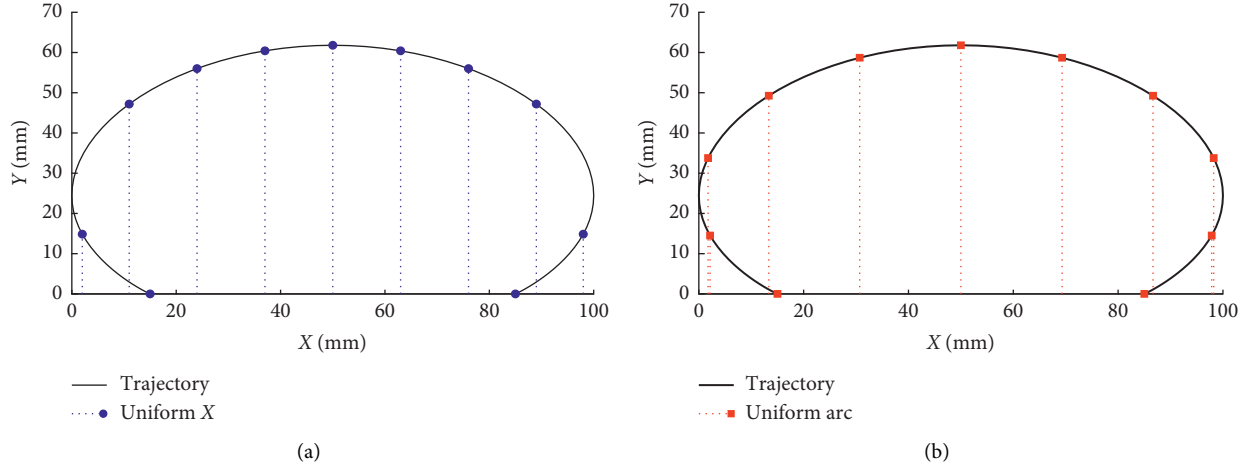


FIGURE 8: Spatial domain trajectory divisions. (a) Uniform X. (b) Uniform arc.

TABLE 3: Corresponding trajectory points of uniform X.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.1999	2	14.8542
3	0.3596	11	47.1906
4	0.4144	24	56.0035
5	0.4589	37	60.4216
6	0.5000	50	61.8000
7	0.5411	63	60.4216
8	0.5856	76	56.0035
9	0.6404	89	47.1906
10	0.8001	98	14.8542
11	1	85	0

TABLE 4: Corresponding trajectory points of uniform arc.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.1980	2.1459	14.5252
3	0.2940	1.7623	33.7796
4	0.3710	13.3457	49.2685
5	0.4380	30.6892	58.7008
6	0.5000	50	61.8000
7	0.5620	69.3108	58.7008
8	0.6290	86.6543	49.2685
9	0.7060	98.2377	33.7796
10	0.8020	97.8541	14.5252
11	1	85	0

space trajectory planning. Finally, the overall joint angle curves are shown in Figure 9.

In Figure 9, the black solid line is the original joint trajectory corresponding to the original design Cartesian trajectory, HFE joint is the hip flexion/extension joint, and KFE joint is the knee flexion/extension joint [35]. By comparing the joint trajectories obtained from the two trajectory points' distribution in Figure 9, it can be seen that

the joint trajectories generated by the two methods are basically close to the original joint trajectory, and the trend is the same. However, there are some errors in details, especially in the beginning and end areas of the trajectories. It can be seen from the local magnifying figure at the right bottom that the joint trajectory obtained by uniform arc division is closer to the original joint trajectory than that by uniform X division. On the other hand, it can be found from Figure 8 that the distribution of trajectory points by uniform arc is more dense in the starting and ending areas. So, there is a guessing that enhancing the points density in the starting and ending areas may help to increase the accuracy. Accordingly, distribution of the trajectory points needs to be further adjusted.

3.2. Trajectory Division in Temporal Domain. Owing to the bionic trajectory is a set of parameter equations about time t , it can directly divide the time to get the coordinate values of X and Y at the same time values, and then the corresponding trajectory points can be obtained and the trajectory can be divided. Referring to the current common methods of parametric function interpolation, the choosing of interpolation points in segmented spline interpolation is usually by the methods of uniform parameterization, chord length, centripetal model, etc. [36]. On the other hand, in the interpolation of higher-order polynomial functions, in order to avoid Runge phenomenon, Chebyshev points are often used as interpolation points [23]. Therefore, referring to the above methods, the time t is divided into two modes: uniform time and Chebyshev nodes. Accordingly, the corresponding coordinate values are solved, and then the inverse kinematics is used to obtain the interpolation points and the joint space trajectory planning is performed, so that the coordinate values corresponding to the isochronous points are shown in Table 5, and its distribution in the trajectory is shown in Figure 10(a).

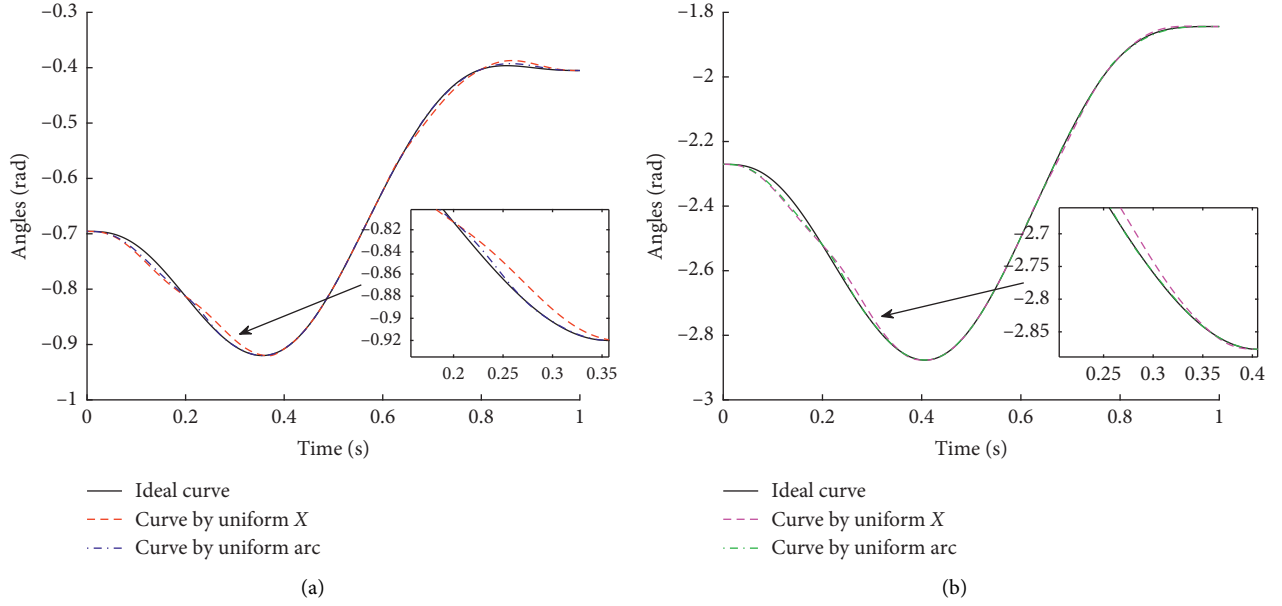


FIGURE 9: Joint trajectory effect based on spatial divisions. (a) HFE joint trajectory. (b) KFE joint trajectory.

TABLE 5: Corresponding trajectory points of uniform time.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.1	11.5056	2.5247
3	0.2	1.9952	14.8716
4	0.3	2.2816	35.0628
5	0.4	20.1888	54.0017
6	0.5	50	61.8000
7	0.6	79.8112	54.0017
8	0.7	97.7184	35.0628
9	0.8	98.0048	14.8716
10	0.9	88.4944	2.5247
11	1	85	0

Chebyshev points are generated by Chebyshev polynomials, which are divided into Chebyshev polynomial of the first class and Chebyshev polynomials of the second class. Chebyshev points can be obtained by finding the zero point or the extreme point of Chebyshev polynomials. Among them, Chebyshev polynomial of the first class can be obtained by [37]

$$\begin{aligned}
 T_0(x) &= 1, \\
 T_1(x) &= x, \\
 T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x),
 \end{aligned} \tag{12}$$

where $n=0, 1, 2, \dots, N$. Equation (12) has $n+1$ extreme points (including endpoints) in $[-1, 1]$, which can be obtained by

$$x_k = \cos\left(\frac{k\pi}{n}\right), \tag{13}$$

where $k=0, 1, 2, \dots, n$. Taking $n=10$, eleven Chebyshev points can be obtained to divide time, thereby generating 11

corresponding trajectory points as shown in Table 6, and the schematic diagram of trajectory points distributions is as shown in Figure 10(b).

In Figure 10, it can be found that the distribution of the trajectory points generated by the two modes is more dense in the beginning and end segments, while the distribution of the trajectory points in the middle segment is sparse. Among them, the distribution based on Chebyshev nodes is the most obvious. The joint trajectory effects of two modes are shown in Figure 11.

By comparing Figure 9 with Figure 11, it can be seen that the effect of joint trajectory generated based on temporal domain division is better than the joint trajectory divided by spatial domain on the whole and is closer to the original joint trajectory. Compared with the two partition methods in Figure 11, it can be found that the trajectory generated by the uniform time distribution is closer to the original joint trajectory.

4. Interpolation Points Optimization

4.1. Optimization Process Analysis. On the basis of the above analysis, in order to explore the optimal distribution of trajectory points, the intelligent optimization algorithms are used to optimize the positions to obtain the best trajectory reducibility. Among many intelligent optimization algorithms [38–40], the PSO [41] and GA [42] are chosen due to their fast convergence and robustness. The primary purpose of optimization is to make the generated trajectory closer to the original design trajectory. Therefore, taking the errors between the original Cartesian trajectory and the actual trajectory as the optimization goal, the fitness function of the PSO algorithm is constructed as shown in equation (14), and the fitness function of GA as shown in equation (15):

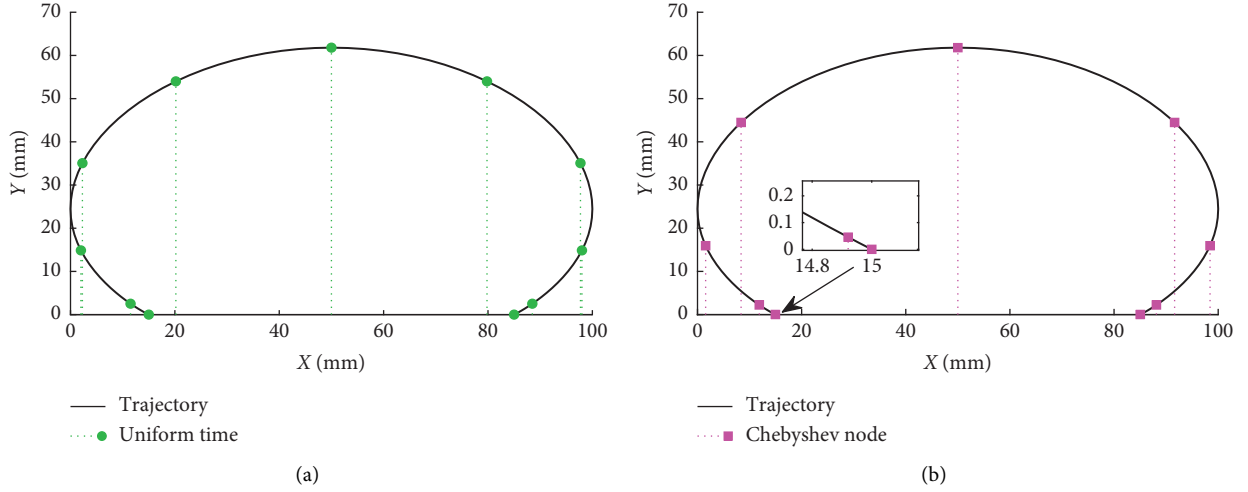


FIGURE 10: Temporal domain trajectory divisions. (a) Uniform time. (b) Chebyshev nodes.

TABLE 6: Corresponding trajectory points of Chebyshev nodes.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.0245	14.9209	0.0455
3	0.0955	11.8685	2.2273
4	0.2061	1.5635	15.9486
5	0.3455	8.3703	44.4837
6	0.5	50	61.8000
7	0.6545	91.6297	44.4837
8	0.7939	98.4365	15.9486
9	0.9045	88.1315	2.2273
10	0.9755	85.0791	0.0455
11	1	85	0

$$f_p = \frac{T}{n} \sum_{i=1}^n \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}, \quad (14)$$

$$f_g = \frac{1}{(T/n) \sum_{i=1}^n \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}, \quad (15)$$

where \hat{x}_i and \hat{y}_i are the Cartesian space coordinate values obtained from the positive kinematics solution of the joint trajectory, x_i and y_i are the original Cartesian trajectory coordinate values corresponding to \hat{x}_i and \hat{y}_i , T is the total motion time, and n is number of sampling points.

Let time t be the independent variable, the random points' distributions as the initial positions. By continuously updating the trajectory points' positions and assessing the fitness, the distributions of trajectory points are gradually optimized until the convergence conditions are met. This process can be represented by the flowchart as shown in Figure 12 below. Among them, (a) is the PSO flowchart and (b) is the GA flowchart.

In the process as shown in Figure 12, in order to obtain the current individual's fitness, the positive kinematics calculation needs to be performed after the joint space trajectory planning to obtain the actual foot-end motion

trajectory. And then the error analysis based on equations (14) and (15) is executed to return the fitness value by comparing with the original Cartesian trajectory. In this process, there are multiple kinematic solving steps, so the combination of PSO or GA function with Simulink is used for optimization. As shown in Figure 13, the fitness calculation system is established through Simulink, and the PSO or GA function is responsible for calling the Simulink model to calculate the fitness, and then the current individual positions are adjusted to further optimize based on the returned fitness value obtained by the Simulink model.

In Figure 13, "Points_in" is used to call the multidimensional individual positions which are the global variables in the workspace generated by the PSO function, and then they are put into "Interpolation_model" to calculate the boundary conditions of each interpolation point. Furthermore, the corresponding piecewise quintic polynomial interpolation function is generated to perform joint space trajectory planning. The "Forward-kinematics" are used to obtain the actual foot-end trajectory by forward kinematics solving of the generated joint trajectory. The actual foot-end trajectory and the original trajectory generated in "Trajectory_function" are put into "Error_analysis" to analyze the error between them by equations (14) and (15) and obtain the fitness. Finally, the fitness is returned to the PSO or GA function through "Out_fitness" for next iteration.

4.2. Optimization Results Analysis. Through the above PSO and GA method, the final obtained distributions of the trajectory points are shown in Tables 7 and 8. From these two tables, we can find that the distributions obtained by the two methods are relatively close, and both are further optimized on the basis of the isochronous distributions.

In order to more comprehensively evaluate the closeness between the actual trajectory and the design trajectory, the integrated absolute error (IAE) and integrated square error (ISDE) [43] are used as the performance indexes. They are shown as follows:

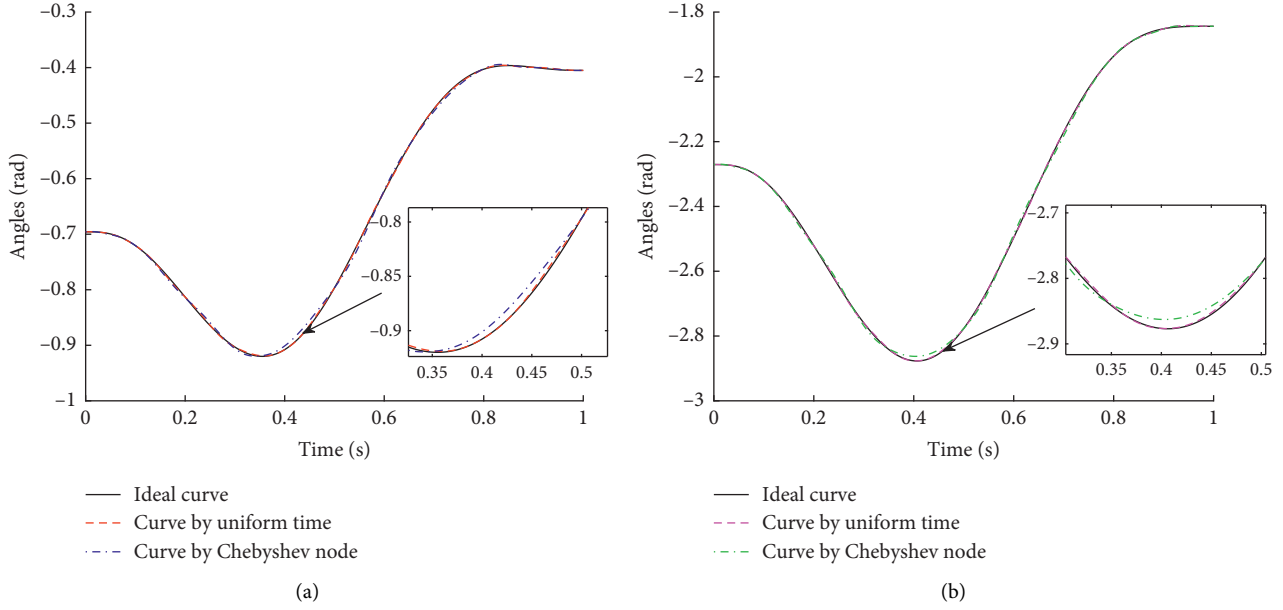


FIGURE 11: Joint trajectory effect based on temporal divisions. (a) HFE joint trajectory. (b) KFE joint trajectory.

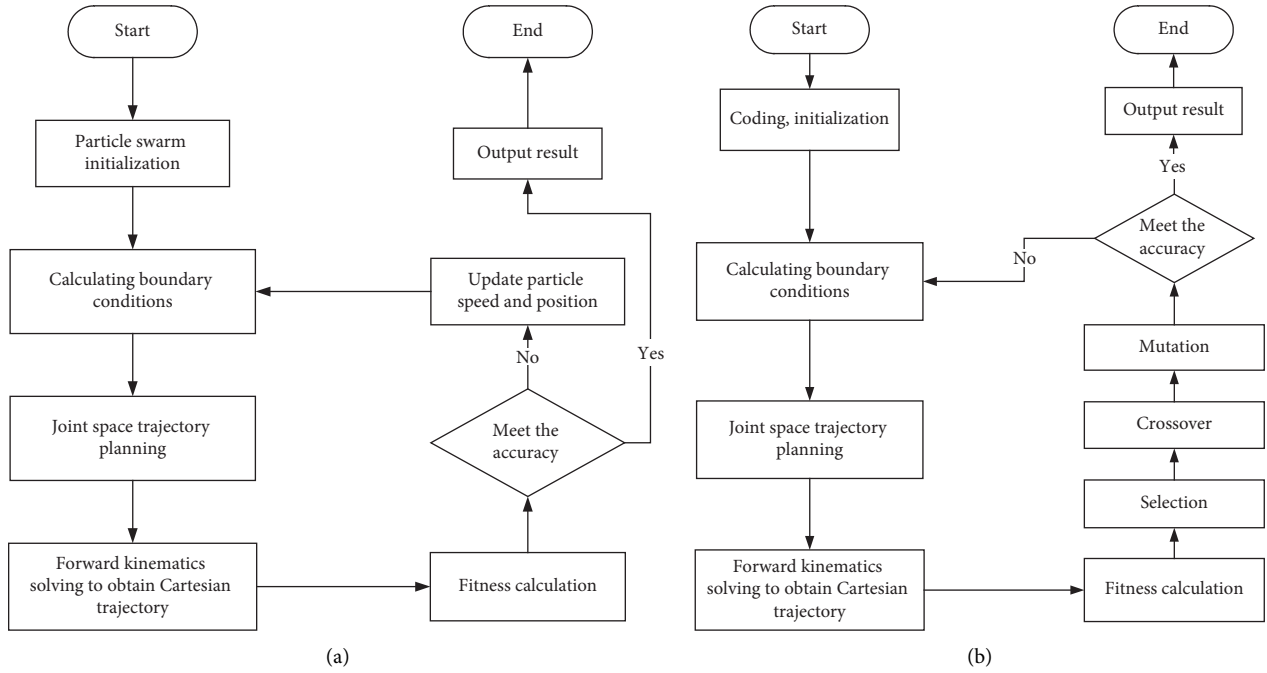


FIGURE 12: Intelligent optimization algorithms flowchart for point distributions. (a) PSO flowchart. (b) GA flowchart.

$$IAE = \int |e(t)|dt, \quad (16)$$

$$ISDE = \int (e(t) - e(0))^2 dt, \quad (17)$$

where $e(t) = \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}$ and $e(0)$ represents the average error. The distributions of relevant trajectory points in Section 2 are compared, and the different performance index values of these distribution modes are shown in Table 9.

According to the analysis of the data in Table 9, the errors of the trajectory generated by dividing time are overall fewer than those generated by dividing space, which further verifies the relevant conclusions in Section 2. When comparing the index values of the optimized points and the uniform time, it can be found that the results are close. The optimal distributions improve by 1.8% to 2.0% of IAE and 13.1% to 17.8% of ISDE compared with the uniform time distributions. Therefore, in the case of less precision requirements, the use of isochronous distribution of trajectory points has been able to achieve better trajectory reduction

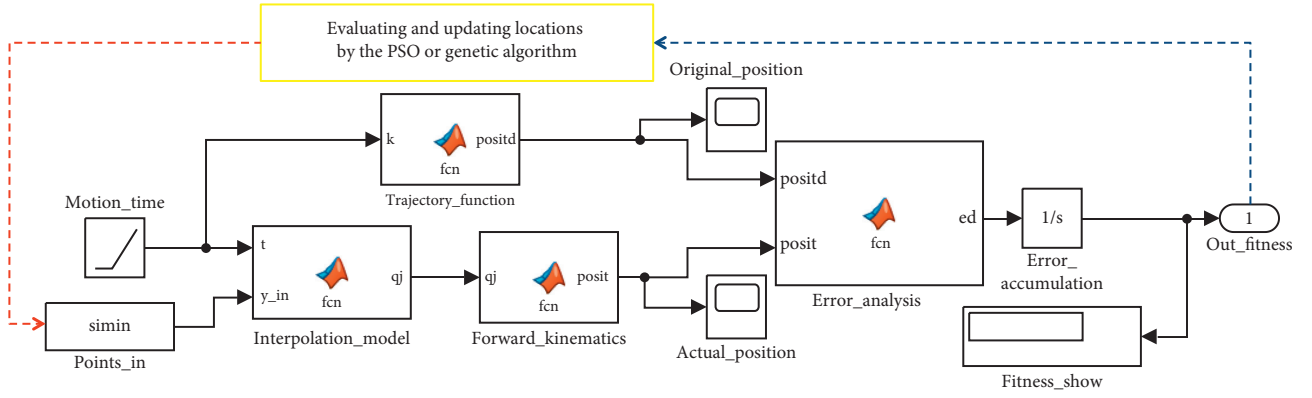


FIGURE 13: Model of fitness calculation system.

TABLE 7: Corresponding trajectory points obtained by PSO.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.1024	11.3051	2.6924
3	0.2048	1.6519	15.7165
4	0.3013	2.4030	35.3404
5	0.4012	20.4961	54.1780
6	0.5027	50.8640	61.7940
7	0.6001	79.8367	53.9869
8	0.6981	97.5399	35.4684
9	0.7933	98.4766	16.0561
10	0.8969	88.7543	2.7425
11	1	85	0

TABLE 8: Corresponding trajectory points obtained by GA.

Point number	Time (s)	X position (mm)	Y position (mm)
1	0	15	0
2	0.1055	11.0393	2.9184
3	0.2031	1.7704	15.4151
4	0.3005	2.3279	35.1696
5	0.3996	20.0868	53.9425
6	0.5006	50.1920	61.7997
7	0.5999	79.7857	54.0164
8	0.6979	97.5207	35.5111
9	0.7922	98.5489	16.2541
10	0.8922	89.1626	3.0930
11	1	85	0

TABLE 9: Performance index values of multiple distribution types.

Distribution mode of trajectory points	IAE	ISDE
Uniform X	1.3860	1.6320
Uniform arc	0.6194	0.6010
Uniform time	0.2560	0.0214
Chebyshev nodes	0.7830	0.3606
PSO	0.2514	0.0186
GA	0.2505	0.0176

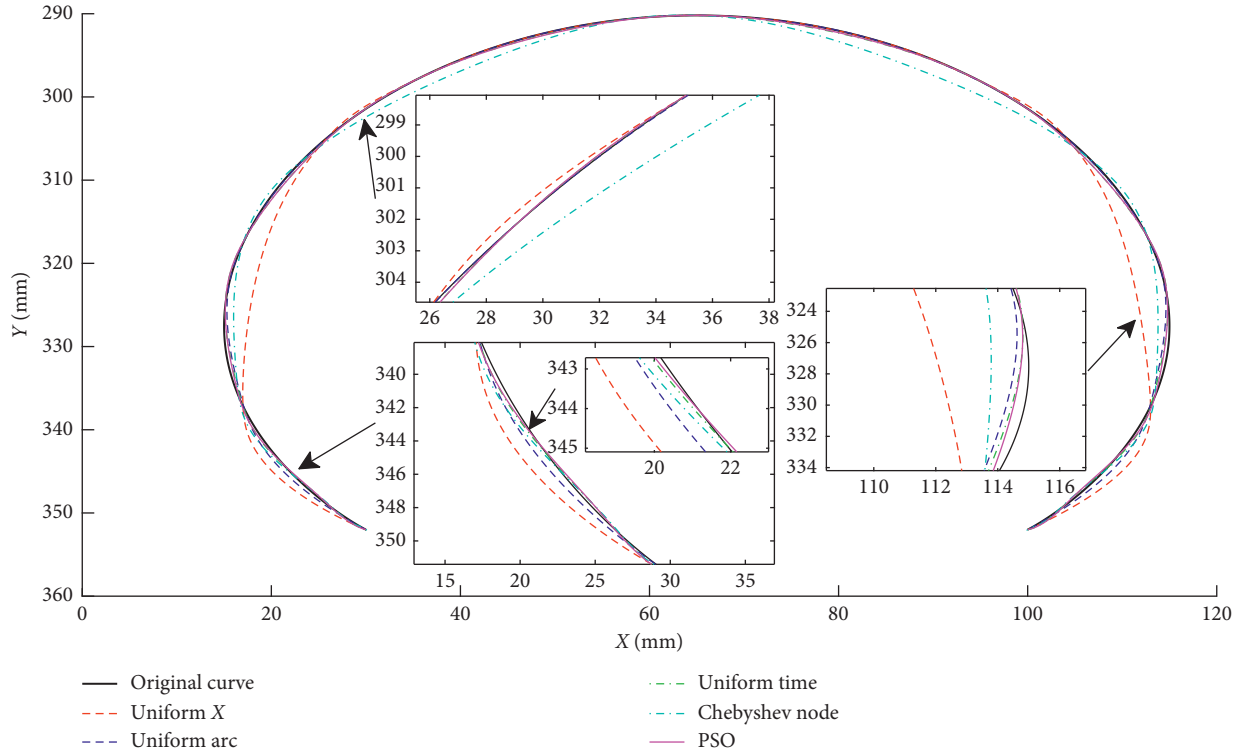


FIGURE 14: Comparison of different trajectory restoration effects.

effect. Moreover, the computation is simple by this method. In addition, if the accuracy needs to be further improved, the PSO or GA method in here can be used to further optimize the point distributions.

5. Simulation Analysis and Prototype Experiment

Based on the above analysis conclusions, a virtual prototype simulation and a prototype experiment are performed to further verify the above optimization results.

By building a virtual prototype associated simulation platform, the motion simulation of a virtual prototype is performed. In this process, the input and output interfaces of each platform are set up for driving and feedback. The motion time and the point distributions are inputted in Simulink to plan the joint space trajectory, and the corresponding joint angle output is got. The joint angle output by Simulink is used as the input of virtual prototype for joint driving and locomotion. The comparison between the actual foot-end trajectory generated by several distribution modes and the original Cartesian trajectory is shown in Figure 14.

In Figure 14, because the points obtained by PSO and genetic algorithm are relatively close, this will cause overlap. After considering the accuracy and convergence speed of the two methods, the point distributions obtained

by PSO are analyzed owing to this method having a faster convergence speed, and the accuracy is close to genetic algorithm. Thus, the simulation of the motion process and the trajectory curve of the foot-end are obtained, which are shown in Figure 15.

In Figure 15, the movement process of the foot-end in a trajectory cycle is simulated. From the partially enlarged figure of each picture and Figure 14, it can be seen that the motion trajectory of the foot-end is consistent with the designed trajectory curve. Then, the foot-end motion characteristics are analyzed, the obtained speed curve of the foot-end is shown in Figure 16(a), and the acceleration curve is shown in Figure 16(b).

It can be seen that the foot-end velocity curve of the virtual prototype generated by the joint trajectory planning is basically consistent with the designed velocity curve, and there is a small fluctuation around the designed speed curve and it is relatively smooth in general comparing Figure 16 with Figures 4 and 5. The acceleration curve of the foot-end is consistent with the designed acceleration curve in the overall trend, but the details are different. Although the acceleration curve of the foot is continuous, the fluctuation range is obvious and the smoothness is poor. It will cause the vibration and resonance of the machine during the actual movement [44, 45]. How to optimize the boundary conditions of the

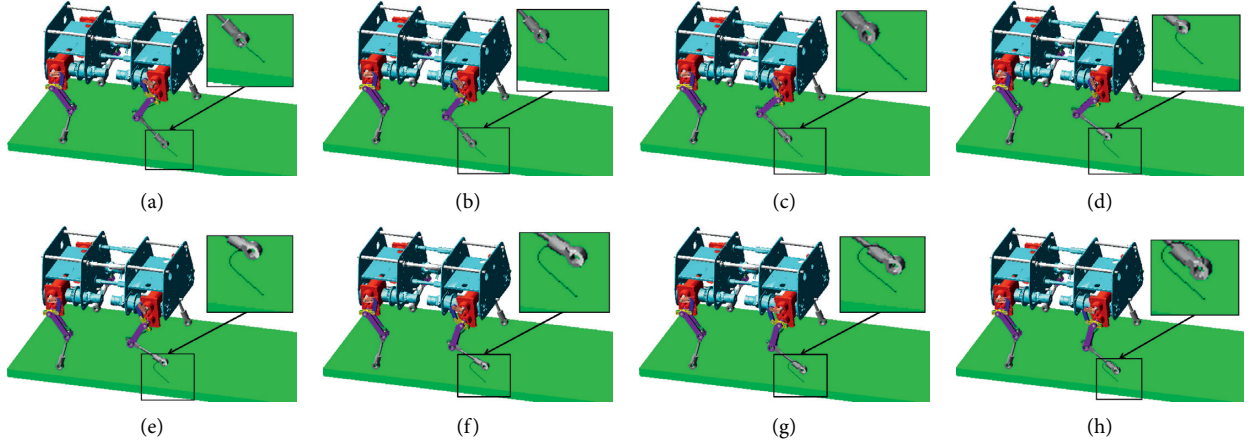


FIGURE 15: Virtual prototype motion process using the point distributions obtained by PSO algorithm. (a) Start. (b) 1/7 period. (c) 2/7 period. (d) 3/7 period. (e) 4/7 period. (f) 5/7 period. (g) 6/7 period. (h) 1 period.

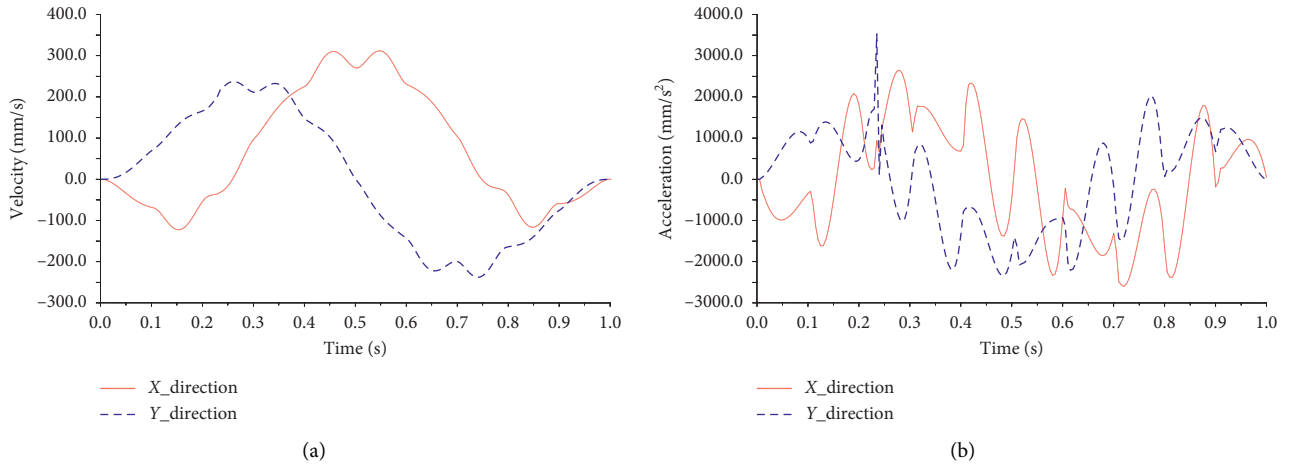


FIGURE 16: Foot-end motion characteristics of virtual prototype. (a) Foot-end speed curve. (b) Foot-end acceleration curve.

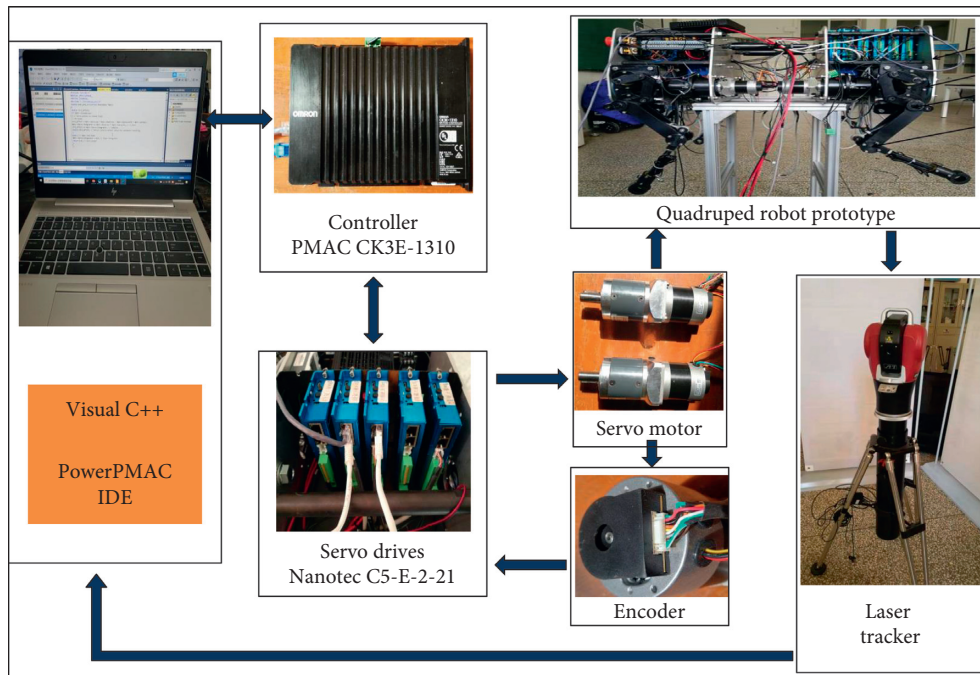


FIGURE 17: Experimental platform for bionic trajectory testing.

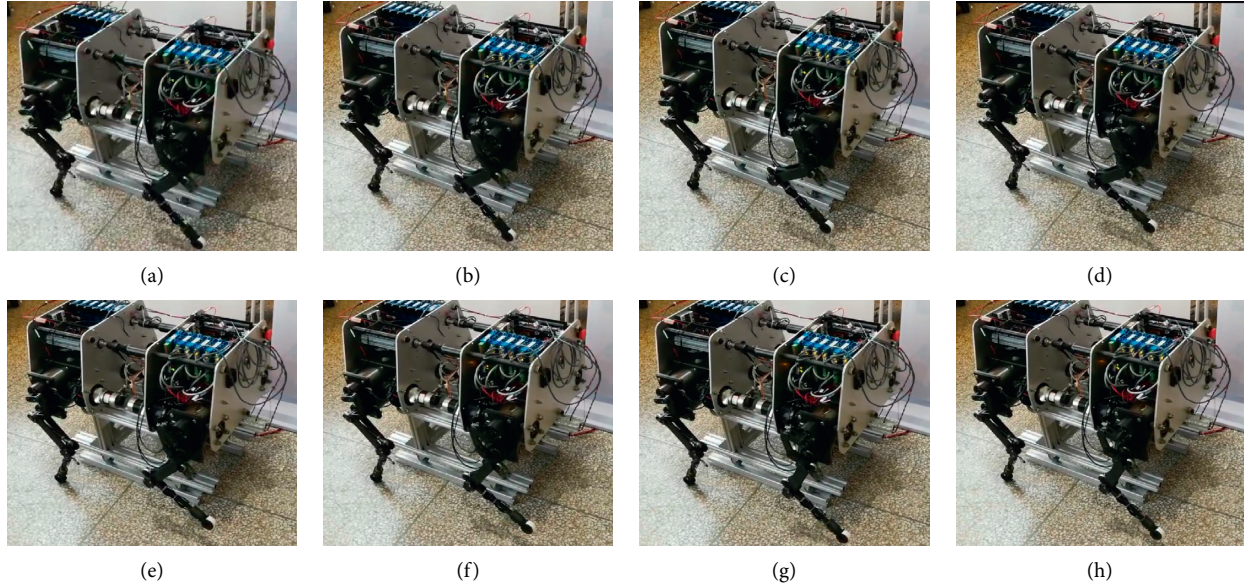


FIGURE 18: Movement process of foot-end of prototype. (a) Start. (b) 1/7 period. (c) 2/7 period. (d) 3/7 period. (e) 4/7 period. (f) 5/7 period. (g) 6/7 period. (h) 1 period.

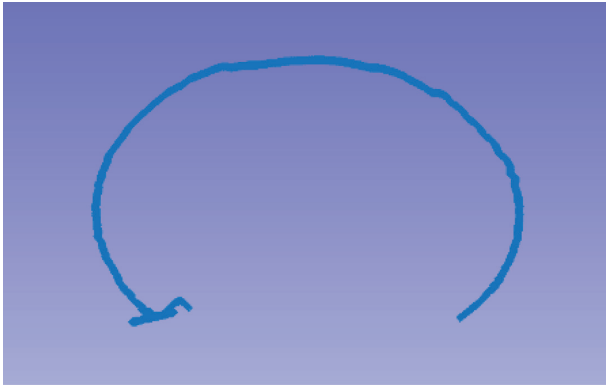


FIGURE 19: Laser tracker sampling trajectory by prototype experiment.

interpolation points is the key to solve the above acceleration unsmoothness problem, and it is also the topic that may need to be researched after this paper.

The experimental platform of quadruped robot prototype is shown in Figure 17, and the interpolation points obtained by PSO algorithm are used as the experimental objects for single-leg motion experiment.

In Figure 17, the bus controller is used to control the motor through Ethernet communication, and the laser tracker is used to track the motion trajectory in real time. The actual motion process of the prototype is shown in Figure 18.

During the experiment, the movement was at a steady performance and achieved a good practical effect. However, due to the gap in the keyway of the leg joint of the prototype, some structural jitter occurs when it is magnified to the foot-end. But, for industrial robots, the effects may be better due to the well assembly accuracy. The sampling effect obtained by the laser tracker is shown in Figure 19. Compared with the design trajectory in

Figure 3, it can be seen that the optimized interpolation points can better restore the original trajectory in the actual foot-end movement.

6. Conclusion

- (1) The Cartesian space trajectory planning combined with the joint space trajectory planning is implemented in the foot-end trajectory planning of quadruped robots. And the piecewise quintic polynomials for trajectory generation are used for the two methods, which have good coordination and consistency.
- (2) The position distribution of interpolation points used in joint space trajectory is explored. The original trajectory is divided into four distribution types: uniform X , uniform arc, uniform time, and Chebyshev nodes. Through analysis and comparison, it was found that the trajectory restoration effect based on temporal division was better than that based on spatial division. Among the four methods, the trajectory reduction degree based on uniform time partition was the highest.
- (3) On the basis of the above analysis, the PSO and GA are used to optimize the distribution of interpolation points, and the points distribution with higher reduction degree is obtained, which is greatly improved compared with the above four methods. Therefore, in the case that accuracy demand is not particularly strict, the isochronous distribution mode can be utilized since it already has high reduction degree and calculation convenience. In addition, if the accuracy needs to be further improved, the PSO or GA method in here can be used to further optimize the point distributions.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 51965029, 61873115, and 51565021) and National Key Research and Development Plan Project (No. 2017YFC1702503).

References

- [1] M. Hutter, C. Gehring, M. A. Höpflinger, M. Blösch, and R. Siegwart, "Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1427–1440, 2014.
- [2] H. Li, A. Shi, and Z. Dai, "A trajectory planning method for sprawling robot inspired by a trotting animal," *Journal of Mechanical Science and Technology*, vol. 31, no. 1, pp. 327–334, 2017.
- [3] J. Jahanpour, M. Motallebi, and M. Porghoveh, "A novel trajectory planning scheme for parallel machining robots enhanced with NURBS curves," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 2, pp. 257–275, 2016.
- [4] M. Mihelj, "Trajectory planning," in *Robotics*, pp. 123–132, Springer, Cham, 2019.
- [5] T. Kagawa, H. Ishikawa, T. Kato, C. Sung, and Y. Uno, "Optimization-based motion planning in joint space for walking assistance with wearable robot," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 415–424, 2015.
- [6] Y. Sakakibara, K. Kan, Y. Hosoda, M. Hattori, and M. Fujie, "Foot trajectory for a quadruped walking machine," in *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, towards a New Frontier of Applications*, Ibaraki, Japan, 1990.
- [7] L. Wang, J. Wang, S. Wang, and Y. He, "Strategy of foot trajectory generation for hydraulic quadruped robots gait planning," *Journal of Mechanical Engineering*, vol. 49, no. 1, pp. 39–44, 2013.
- [8] C. Chevallereau, A. Formal'sky, and B. Perrin, "Control of a walking robot with feet following a reference trajectory derived from ballistic motion," in *Proceedings of the International Conference on Robotics and Automation*, Albuquerque, NM, USA, 1997.
- [9] P. R. Vundavilli and D. K. Pratihari, "Dynamically balanced optimal gaits of a ditch-crossing biped robot," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 349–361, 2010.
- [10] K. Y. Kim and J. H. Park, "Ellipse-based leg-trajectory generation for galloping quadruped robots," *Journal of Mechanical Science and Technology*, vol. 22, no. 11, pp. 2099–2106, 2008.
- [11] C. Semini, V. Barasuol, T. Boaventura et al., "Towards versatile legged robots through active impedance control," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 2015.
- [12] X. Rong, Y. Li, J. Ruan, and B. Li, "Design and simulation for a hydraulic actuated quadruped robot," *Journal of Mechanical Science and Technology*, vol. 26, no. 4, pp. 1171–1177, 2012.
- [13] Z. Hua, X. Rong, Y. Li, H. Chai, and S. Zhang, "Active compliance control on the hydraulic quadruped robot with passive compliant servo actuator," *IEEE Access*, vol. 7, pp. 163449–163460, 2019.
- [14] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1417–1445, 2014.
- [15] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, Pearson Education India, Bengaluru, India, 3rd edition, 2009.
- [16] C.-S. Lin and P.-R. Chang, "Joint trajectories of mechanical manipulators for Cartesian path approximation," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, no. 6, pp. 1094–1102, 1983.
- [17] P. I. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics & Automation Magazine*, vol. 3, no. 1, pp. 24–32, 1996.
- [18] R. Dehghani, A. Fattah, and E. Abedi, "Cyclic gait planning and control of a five-link biped robot with four actuators during single support and double support phases," *Multibody System Dynamics*, vol. 33, no. 4, pp. 389–411, 2015.
- [19] A. Gasparetto and V. Zanotto, "A technique for time-jerk optimal planning of robot trajectories," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 415–426, 2008.
- [20] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 309–317, 2013.
- [21] J. Zhong, M. Luo, J. Fan, and J. Zhao, "Trajectory planning of an intermittent jumping quadruped robot with variable redundant and underactuated joints," *Complexity*, vol. 2018, Article ID 5369427, 14 pages, 2018.
- [22] M. Wang, J. Luo, and U. Walter, "Trajectory planning of free-floating space robot using particle swarm optimization (PSO)," *Acta Astronautica*, vol. 112, pp. 77–88, 2015.
- [23] D. Chen, L. Wang, and J. Zhang, "A homogeneous high precision direct integration based on Chebyshev interpolation," *Vibroengineering PROCEDIA*, vol. 22, pp. 170–175, 2019.
- [24] J. R. Usherwood, "An extension to the collisional model of the energetic cost of support qualitatively explains trotting and the trot-canter transition," *Journal of Experimental Zoology Part A: Ecological and Integrative Physiology*, vol. 333, no. 1, pp. 9–19, 2019.
- [25] H. M. Herr and T. A. McMahon, "A trotting horse model," *The International Journal of Robotics Research*, vol. 19, no. 6, pp. 566–581, 2000.
- [26] A. M. Wilson, J. C. Lowe, K. Roskill, P. E. Hudson, K. A. Golabek, and J. W. McNutt, "Locomotion dynamics of hunting in wild cheetahs," *Nature*, vol. 498, no. 7453, pp. 185–189, 2013.
- [27] M. Hildebrand, "Further studies on locomotion of the cheetah," *Journal of Mammalogy*, vol. 42, no. 1, pp. 84–91, 1961.
- [28] H. Li, *Biomimetic Locomotion Control Theories and Methods of Quadruped Robot*, Beijing Institute of Technology, Beijing, China, 2014.
- [29] J. Chen, H. San, and X. Wu, "Gait regulation of a bionic quadruped robot with antiparallelogram leg based on CPG

- oscillator,” *Complexity*, vol. 2019, Article ID 5491298, 11 pages, 2019.
- [30] J. H. Lee and J. H. Park, “Optimization of postural transition scheme for quadruped robots trotting on various surfaces,” *IEEE Access*, vol. 7, pp. 168126–168140, 2019.
 - [31] J. Hwangbo, J. Lee, A. Dosovitskiy et al., “Learning agile and dynamic motor skills for legged robots,” *Science Robotics*, vol. 4, no. 26, 2019.
 - [32] B. A. Ibrahimoglu, “Lebesgue functions and Lebesgue constants in polynomial interpolation,” *Journal of Inequalities and Applications*, vol. 2016, no. 1, p. 93, 2016.
 - [33] H. Idais, M. Yasin, M. Pasadas, and P. González, “Optimal knots allocation in the cubic and bicubic spline interpolation problems,” *Mathematics and Computers in Simulation*, vol. 164, pp. 131–145, 2019.
 - [34] P. Laube, M. O. Franz, and G. Umlauf, “Learnt knot placement in B-spline curve approximation using support vector machines,” *Computer Aided Geometric Design*, vol. 62, pp. 104–116, 2018.
 - [35] C. Semini, “Design overview of the hydraulic quadruped robots HyQ2Max and HyQ2Centaur,” in *Proceedings of the Fourteenth Scandinavian International Conference On Fluid Power*, Tampere, Finland, 2015.
 - [36] E. T. Y. Lee, “Choosing nodes in parametric curve interpolation,” *Computer-Aided Design*, vol. 21, no. 6, pp. 363–370, 1989.
 - [37] B. K. Ghimire, H. Y. Tian, and A. R. Lamichhane, “Numerical solutions of elliptic partial differential equations using Chebyshev polynomials,” *Computers & Mathematics with Applications*, vol. 72, no. 4, pp. 1042–1054, 2016.
 - [38] S. B. Cheikh and S. Hammadi, “Multi-hop ridematching optimization problem: intelligent chromosome agent-driven approach,” *Expert Systems with Applications*, vol. 62, pp. 161–176, 2016.
 - [39] Q. Chen, H. Shi, and M. Sun, “Echo state network-based backstepping adaptive iterative learning control for strict-feedback systems: an error-tracking approach,” *IEEE Transactions on Cybernetics*, vol. 50, no. 7, pp. 3009–3022, 2020.
 - [40] J. Na, B. Jing, Y. Huang, G. Gao, and C. Zhang, “Unknown system dynamics estimator for motion control of nonlinear robotic systems,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3850–3859, 2020.
 - [41] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of the ICNN’95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, Perth, Australia, August 2002.
 - [42] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, Cambridge, UK, 1st edition, 1992.
 - [43] S. Wang, L. Tao, Q. Chen, J. Na, and X. Ren, “Usde-based sliding mode control for servo mechanisms with unknown system dynamics,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 1056–1066, 2020.
 - [44] A. Valente, S. Baraldo, and E. Carpanzano, “Smooth trajectory generation for industrial robots performing high precision assembly processes,” *CIRP Annals*, vol. 66, no. 1, pp. 17–20, 2017.
 - [45] Q. Chen, S. Xie, M. Sun, and X. He, “Adaptive nonsingular fixed-time attitude stabilization of uncertain spacecraft,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 2937–2950, 2018.