Design of Fuzzy Gait Control Algorithm for Multi-legged Hydraulic Robot

He Zhu, Hongchao Xing, Jinfu Zhu, Ping Zhang

Abstract—**In the traditional robot gait control algorithm, genetic algorithm is utilized to optimize the CPG control parameters, but the local optimal problem is not considered, thus reducing the gait control effect. Thus, the fuzzy gait control algorithm of multi-legged hydraulic robot is designed. BP neural network is used to reduce the output error. Combined with the fuzzy control process, the autonomous navigation closed-loop control algorithm of fuzzy neural network is created to control the gait of the robot. On this basis, the static stability margin is used to maintain the robot's gait statically stable. The research outcomes indicate that the designed algorithm makes the robot successfully avoid obstacles in different environments. During the movement, the joint movement velocity changes smoothly without mutation, the motion error is only 0.1 mm, the dynamic response is fast, and the stability is high, indicating that the algorithm can achieve high-precision, high-efficiency robot gait control with good stability.**

Index Terms—**fuzzy control, gait, hydraulic robot, multi-legged machine, neural network, static stability margin**

I. Introduction

With the sustained advancement of science and technology, the development of the robotic intelligent industry has made many breakthroughs. According to the way the robot walks, the robot can be generally separated into a wheeled robot and a multi-legged robot [1]. As a newcomer in the field of robot research and development, multi-legged hydraulic robots (MLHR) have the advantages of strong carrying capacity, good stability and remarkable adaptability in non-structural environments [2]. The superiority of a MLHR is mainly manifested by walking on non-structural ground [3]. The the point contact form is adopted when the MLHR contacts the ground, the highly flexible joint design and the foot end structure design of multiple support points can adjust the height of the center of gravity in real-time. Therefore, compared to wheeled robots, these characteristics make it more environmentally adaptable and stable in irregular terrain. The gait of a MLHR includes two types: regular gait under flat terrain and free gait under rough terrain. It can change the gait pattern and leg posture of its travel according to the different terrains of the environment, thus ensuring the flexibility and stability of walking in an unstructured environment.

Gait refers to the walking mode of the walking structure [4], which is the sequence and manner of lifting and releasing the leg when the walking structure is walking. The hexapod robot's gait planning is to study the gait control issue when the robot is walking, and has become one of the most critical technologies for robot research. The hexapod robot pose motion refers to the positional motion and posture motion of the platform without relative movement between the foot and the ground. In traditional gait control algorithm of MLHR based on CPG control model [5], for the problem that the parameters of CPG control model are difficult to be fixed, genetic algorithm is used to optimize it and realize CPG based gait planning, and it did not consider the "too close" problem in the gait control process, with the problem of sudden changes in velocity and low accuracy in gait control. Therefore, the fuzzy gait control (FGC) algorithm of the MLHR is designed, combined with the static stability margin (SSM) to achieve uniform, high-precision and stable gait control.

II. FGC algorithm for MLHR

A. *MLHR control algorithm based on fuzzy neural network (FNN)*

In this paper, the MLHR control system is studied, and a closed-loop control algorithm based on FNN is designed.

BP neural network

The back propagation (BP) neural network algorithm calculates the weight threshold matrix parameters of the network model based on the actual input and output data. It contains two parts, which are the forward propagation of the signal and the BP of the error [6]. The algorithm flow chart is shown in Figure 1.

Fuzzy Control (FC)

The principle of FC is to perform corresponding fuzzy logic reasoning on events through experience and rules [7], and use fuzzy language to describe, which solves the problem that accurate mathematical model is hard to be built with the increase of system complexity [8]. The FC structure is shown in Figure 2.

Manuscript received April 4, 2022; revised June 25, 2023.

The research is supported by: Anshun Power Supply Bureau of Guizhou Power Grid Co., Ltd. 2019 Science and Technology Project: "Research and Development of Climbable Transformer Robots", Project Number: 060400KK52180017.

He Zhu is a vice president of Graduate School from Northeast Electric Power University, Jilin 132000, China. (corresponding author to provide phone: 13985302715; e-mail: zhuhelunwen@163.com).

Hongchao Xing is an undergraduate student of transmission engineering from Northeast Electric Power University, Jilin 132000, China. (e-mail: xinghc9612@163.com).

Jinfu Zhu is an undergraduate student of transmission engineering from Northeast Electric Power University, Jilin 132000, China. (e-mail: zhujinfu1234567@163.com).

Ping Zhang is an undergraduate student of transmission engineering from Northeast Electric Power University, Jilin 132000, China. (e-mail: zhang815223701@163.com).

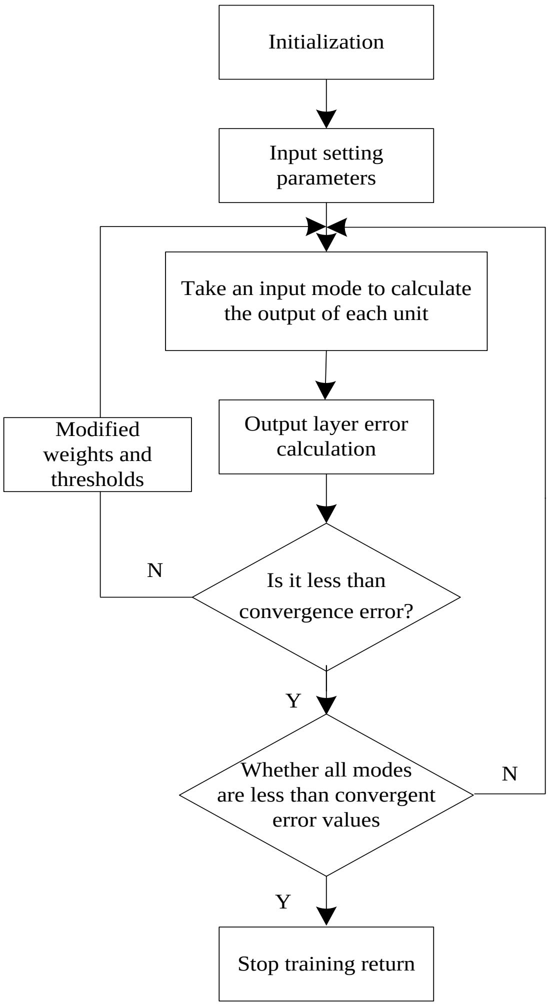

Initialization
Input setting parameters
Take an input mode to calculate the output of each unit
Output layer error calculation
Modified weights and thresholds
Is it less than convergence error?
N
Y
Whether all modes are less than convergent error values
N
Y
Stop training return

Fig. 1. BP algorithm flow chart.

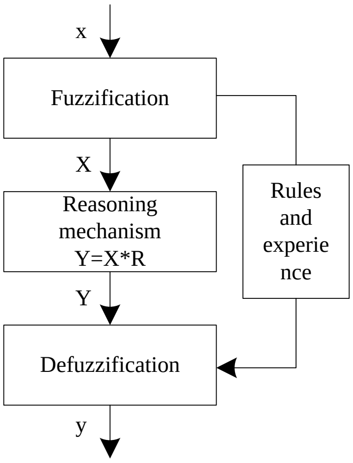


Fig. 2. FC structure.

Where x represents accurate information input, X represents fuzzified information input, y represents accurate information output, Yrepresents fuzzified information output, and relationship R is a conversion relationship function in the inference mechanism, and the role is to determine the output information of FC according to the current input information [9].

The process of FC is mainly three steps: fuzzification, fuzzy reasoning mechanism, and defuzzification calculation [10].

The first step is fuzzification, which is described using a membership function. How to determine the membership function is very important for FC. The membership function reflects the gradual characteristics of the information of things [11], while observing some basic principles.

The fuzzy set of the membership function requires the characteristics of convex blur. In practice, the membership function of triangle or trapezoid is generally used.

The membership function requires maintaining symmetry characteristics and conforming to the logical order and common sense experience.

Each data entered should be guaranteed to belong to at least one and no more than two membership function regions [12].

(4) For the same input data, there will be no case where the two membership functions in which the two membership functions are present at the same time.

The second step is the reasoning mechanism. The key to FC is the equation of the reasoning mechanism. It is designed based on experience and rules.

The third step is to defuzzify, and the output data obtained through the inference mechanism is a fuzzy value. In practical applications, we need to convert the fuzzy value into a certain value to control the actuator.

Design of autonomous navigation closed-loop control algorithm based on FNN

Input and output the vector related definition

Definition 1: The space vector (SV) of input state (1) of MLYR is In .

$$In=\begin{bmatrix} D_1 & D_2 & D_3 & D_4 & D_5 & Tg \end{bmatrix}^T \quad (1)$$

In formula (1), D_1 , D_2 , D_3 , D_4 , and D_5 indicate the obstacle distance information captured by the five angles of the ultrasonic sensor fan scanning (-60°, -30°, 0°, 30°, 60°), and Tg means the angle information between the robot's and the current heading angles. These six quantities represent the six dimensions of the input state's SV.

Definition 2: The output motion's SV of the hexapod robot is Ou .

$$Ou=\begin{bmatrix} V & M \end{bmatrix}^T \quad (2)$$

Where V represents the velocity of the MLHR output by the system, and M expresses the steering angle of the MLHR. These two quantities represent the two dimensions of the output state's SV.

Analysis of robot safety performance

The safety of MLYR walking in the autonomous navigation system has great importance. When the robot is walking close to the edge of the obstacle or the corner, it will lead to the problem of “too close” [13]. Therefore, the hydraulic robot needs to maintain a certain safe distance (SD) from the obstacle to perform the steering bypass behavior. The determination of the SD depends on the two dimensions of the robot's external dimensions and travel velocity (TV) [14]. The SD is labeled as SD , and SD represents the SD of the ultrasonic sensor to detect 5 angles. The relation between SD and the TV V_t is as follows:

$$SD \ge \left(V_t + V_{\max}\right)^* \tau + R \quad (3)$$

In formula (3), $V_{\max}$ represents the max TV of the robot, V_t represents the current TV at time t, τ takes the value λ seconds, and $\lambda(>=1)$ is the safety threshold, which is based on actual demand (the value in this paper is $\lambda=1$). R is the space between the outermost point of the MLHR and the mass center of the machine. The SD changes timely with the change of the robot's TV V_t, avoiding the path redundancy because of a single defined SD [15].

Closed-loop control algorithm based on FNN

To make the control system simpler, it designs a 6-input and 2-output fuzzy BP neural network algorithm. The whole algorithm is divided into 5 layers, as shown in Figure 3.

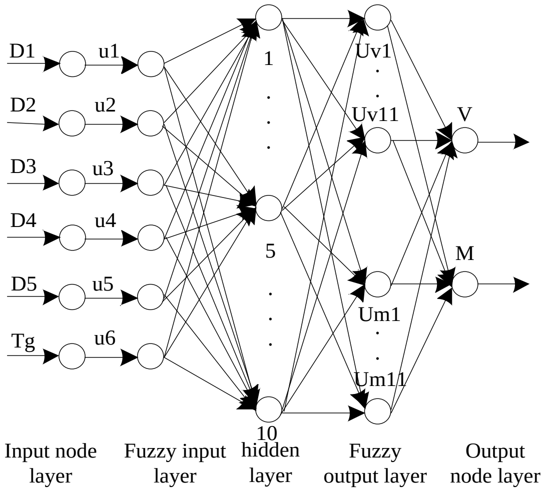


Fig. 3. FNN.

The input node layer is the first one, and the surrounding environment information sensed by the sensor constitutes the input state's SV In (Formula 1). This layer' nodes are directly linked to the constituents D_1, D_2, D_3, D_4, D_5, Tg of the SV x and passed to the next layer. The input vector is fuzzy, where a node denotes a language variable. In this paper, the input distance information D_1, D_2, D_3, D_4, D_5 includes two levels by comparison with the feedback SD: I (within SD) and O (without SD). In addition, this paper divides the input angle quantity Tg into 5 standards (LB, LM, ZO, RM, RB).

The second layer is the fuzzy input layer. The function is to calculate the fuzzy membership function set of the input state's SV In. The membership function is Gaussian function, which is denoted as:

$$\mu_i(x) = \exp\left(\frac{(x-\beta_i)^2}{\delta_i^2}\right)(i=1,2,\mathrm{L}\ ,n) \quad (4)$$

Where β_i and δ_i indicate the center and width of the Gaussian function, n means the amount of input nodes, and x refers to the SV element. The SV In is fuzzed by the membership function, and the fuzzed vector is $U=\left[\mu_1 \ \ \mu_2 \ \ \mathrm{L} \ \ \ \mu_6\right]^T$.

The third one is the hidden layer, which completes the nonlinear mapping from the fuzzy input layer to the fuzzy output layer. According to the 0.618 segmentation selection method and multiple experimental comparisons, the amount of hidden layer nodes is determined to be 10.

The fourth is the fuzzy output layer, and the fuzzy output h_j is obtained by the membership function. Each node in this layer is also processed using a Gaussian membership function. To make the fuzzy output h_j easy to understand and operativities transformed into the domain [-1, 1], which is expressed in formula (5):

$$g_j = \frac{2}{d-c}\left[h_j - \frac{c+d}{2}\right](j=1,2,\mathrm{L}\ ,k) \quad (5)$$

In formula (5), $[c,d]$ means the range of the actual output, k is the amount of nodes in the fuzzy output layer, g_j stands for the variable in the domain [-1, +1], and the fuzzy vector of the fourth layer is $G=\left[\mu_v(c_1) \ \ \mu_v(c_2) \ \ \mathrm{L} \ \ \mu_v(c_{11}) \ \ \mu_m(c_1) \ \ \mu_m(c_2) \ \ \mathrm{L} \ \ \mu_m(c_{11})\right]^T$. Among them, c_1, c_2, ... c_{11} are -1, -0.8, ...1, respectively. The vector G is a column vector containing 22 elements. The first 11 variables are the fuzzy vector $G_v=\left[\mu_v(c_1) \ \ \mu_v(c_2) \ \ \mathrm{L} \ \ \ \mu_v(c_{11})\right]^T$ for the velocity V, and the last 11 variables are the fuzzy vector $G_m=\left[\mu_m(c_1) \ \ \mu_m(c_2) \ \ \mathrm{L} \ \ \ \mu_m(c_{11})\right]^T$ for the steering angle M.

The fifth is the output node layer, which realizes the defuzzification calculation function of the fuzzy output [16]. It adopts the widely used method in industrial control: weighted average method, and the defuzzification formula of output velocity V and the steering angle M is:

$$V=\frac{\sum_1^{11} c_i \mu_v(c_i)}{\sum_1^{11} \mu_v(c_i)} = \frac{(-1)*\mu_v(-1)+(-0.8)*\mu_v(-0.8)\mathrm{L}\ +\mu_v(1)}{\mu_v(-1)+\mu_v(-0.8)\mathrm{L}\ +\mu_v(1)} \quad (6)$$

$$M=\frac{\sum_1^{11} c_i \mu_m(c_i)}{\sum_1^{11} \mu_m(c_i)} = \frac{(-1)*\mu_m(-1)+(-0.8)*\mu_m(-0.8)\mathrm{L}\ +\mu_m(1)}{\mu_m(-1)+\mu_m(-0.8)\mathrm{L}\ +\mu_m(1)} \quad (7)$$

Where V represents the velocity of the MLHR output by the system. In this paper, V will be divided into three levels: Fa, fast (>20cm/s), Lo, slow (<10cm/s) and Ze, zero velocity (0cm/s). M includes five levels. These two quantities represent the two dimensions of the output state's SV Ou, and $\mu(1)$, $\mu(-1)$ and $\mu(-0.8)$ represent the degree of fuzzification, where the values 1, 1 and -0.8 in the brackets indicate positive and negative fuzzification, respectively, and the degree of negative fuzzification of $\mu(-0.8)$ is higher than $\mu(-1)$.

B. Stability Analysis of Robot Walking

The MLHR ensures that the body is in a state of balance and does not fall over when walking. It is called static stability walk [17]. The gait of a MLHR in this paper is static, regardless of the impact of the inertial power of the robot motion on its stability. In 1976, MCGHEE and ISWANDHI proposed the concept of SSM. The SSM is defined as the minimum distance from the center of gravity projection point to each side of the projected supporting polygon [18]. The mathematical expression is:

$$S_{SSM}=\min_{i}^{l_n}\left(d_{iG}\right) \quad (8)$$

Where S_{SSM} is the SSM; l_n is the amount of support feet, and d_{iG} is the distance from the gravity center to the i the edge of the supporting polygon. When $S_{SSM}>0$, the robot is in a stable state; If not, it is unstable. For MLHRs, when the robot is in motion, its supporting polygons always envelop the projection of its center of gravity in the supporting plane, and the gait of the robot is statically stable [19].

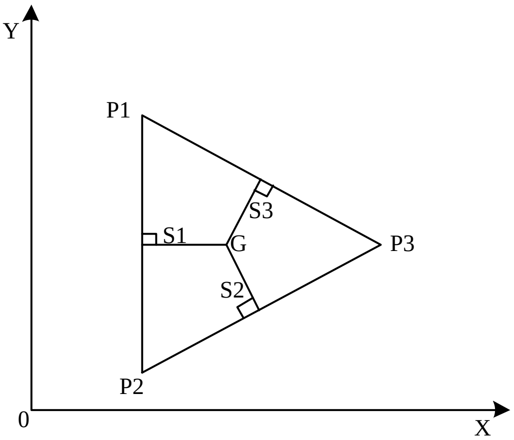


Fig. 4. Center of gravity projection and support polygon.

As shown in Figure 4, the triangular area represents a stable area of the robot in which the foot of the supporting phase is formed in the supporting plane. As long as the projection G of the robot's gravity center of in the supporting plane is always enveloped by the robot, the robot is statically stable. The robot will be unstable when the projection point is beyond the area. It is the distance from the robot's center of gravity projection point G to the robot's supporting triangle boundary, and $S_{SSM}=\min\left\{s_1,s_2,s_3\right\}$ is the stability margin of the robot walking state.

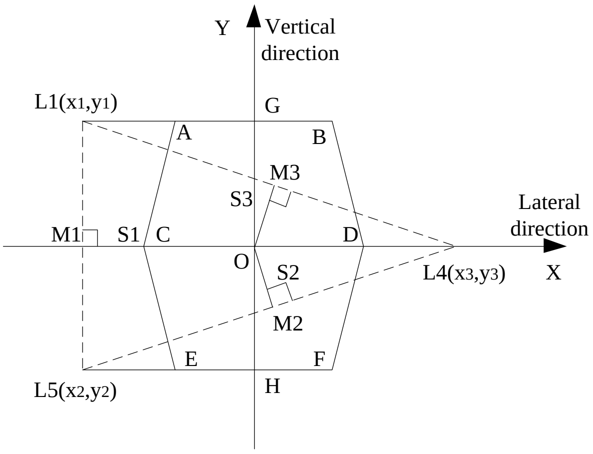


Fig. 5. Relationship between gravity projections of MLHR and its supporting polygon position.

The L1, L4, and L5 points shown in Figure 5 are the support points of a certain robot's three feet on the ground, and the triangle L1L4L5 is a polygon of three support feet in the support plane. The point O is the MLHR's projection of the center of gravity, the X-axis direction is the forward direction of the robot lateral walking, and the Y-axis direction is the direction of the lateral movement of the robot. OM1, OM2, and OM3 are the vertical distances of the robot gravity center from the support plane to the sides of the supporting polygon L1L4L5, and the foot points are M1, M2, M3.

From the coordinates of L1, L5 shown in Figure 5, the slope $k_1=\frac{y_2-y_1}{x_2-x_1}$ of the straight line L1L5 can be obtained, and the straight line L1L5 is expressed as formula (9):

$$y-y_1=\frac{y_2-y_1}{x_2-x_1}\left(x-x_1\right) \quad (9)$$

Since the straight line OM1 and the straight line L1L5 are perpendicular, the slope of OM1 is $k_2=\frac{x_1-x_2}{y_2-y_1}$, then the formula of the straight line OM1:

$$y=\frac{x_1-x_2}{y_2-y_1}x \quad (10)$$

The coordinates of the intersection M1 of the upper two lines can be:

$$M1=\left(\frac{\left(y_2-y_1\right)\left(y_2x_1-x_2y_1\right)}{\left(x_1-x_2\right)^2+\left(y_1-y_2\right)^2},\frac{\left(x_1-x_2\right)\left(y_2x_1-x_2y_1\right)}{\left(x_1-x_2\right)^2+\left(y_1-y_2\right)^2}\right) \quad (11)$$

Using the above method, it can find the coordinates of M2 and M3:

$$M2=\left(\frac{\left(y_3-y_1\right)\left(y_3x_1-x_3y_1\right)}{\left(x_1-x_3\right)^2+\left(y_1-y_3\right)^2},\frac{\left(x_1-x_3\right)\left(y_3x_1-x_3y_1\right)}{\left(x_1-x_3\right)^2+\left(y_1-y_3\right)^2}\right) \quad (12)$$

$$M3=\left(\frac{\left(y_3-y_2\right)\left(y_3x_2-x_3y_2\right)}{\left(x_2-x_3\right)^2+\left(y_2-y_3\right)^2},\frac{\left(x_2-x_3\right)\left(y_3x_2-x_3y_2\right)}{\left(x_2-x_3\right)^2+\left(y_2-y_3\right)^2}\right) \quad (13)$$

It is concluded that the values of the distance s_1,s_2,s_3 from the MLHR gravity center to the three sides of the supporting triangle are:

$$\begin{aligned} s_1&=\frac{\left|y_2x_1-x_2y_1\right|}{\sqrt{\left(x_1-x_2\right)^2+\left(y_1-y_2\right)^2}} \\ s_2&=\frac{\left|y_3x_2-x_3y_2\right|}{\sqrt{\left(x_2-x_3\right)^2+\left(y_2-y_3\right)^2}} \quad (14) \\ s_3&=\frac{\left|y_3x_1-x_3y_1\right|}{\sqrt{\left(x_1-x_3\right)^2+\left(y_1-y_3\right)^2}} \end{aligned}$$

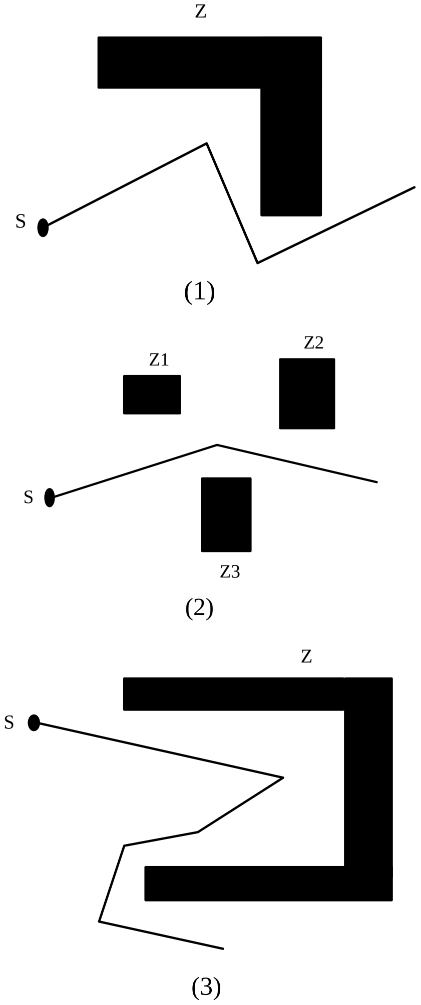
The stability margin of the robot is obtained as $S_{SSM}=\min\limits_{i}^{l_3}\left(s_i\right)=\min\left\{s_1,s_2,s_3\right\}$.

When the robot moves in the lateral direction, it is necessary to ensure that the center of gravity projection point is within the range of the supporting polygon, and the step size theoretically satisfies the relationship: $\frac{s_2}{2}<\left|OL4\right|$, that is $s_2<2\left(\left|OD\right|+d\right)$. The horizontal straight gait [20] is planned in this paper, when the MLHR moves laterally to the right (left), the robot gravity center is always within the range of its supporting triangle, so the stability margin S_{SSM} of the robot's lateral straight walking is greater than zero, the step size satisfies the relationship: $0 < s_2 < 2d$, that is, the projection point of the center of gravity of the robot is always within the range of the supporting triangle, indicating that the horizontal straight process of the robot is statically stable.

III. Results

A. *Obstacle avoidance experiment*

To exam the function of the FGC algorithm of the MLHR designed in this paper, a MLHR of a certain model is taken as the experimental object, and the robot's obstacle avoidance experiment is carried out in the national key laboratory by the proposed algorithm, shown in Figure 6.



(1)

(2)

(3)

Fig. 6. Experimental results of obstacle avoidance for MLHR.

B. *Functional test experiment*

To validate the performance of the proposed algorithm, the algorithm is applied to a MLHR control system of a certain model. After the system is successfully built, it is functionally tested. The test results of actual application function are indicated in Table 1.

Analysis of Table 1 can be obtained that the expected results of the functional test cases of the algorithm are conform to the actual test results, indicating that the proposed algorithm can effectively perform the FGC of MLHR.

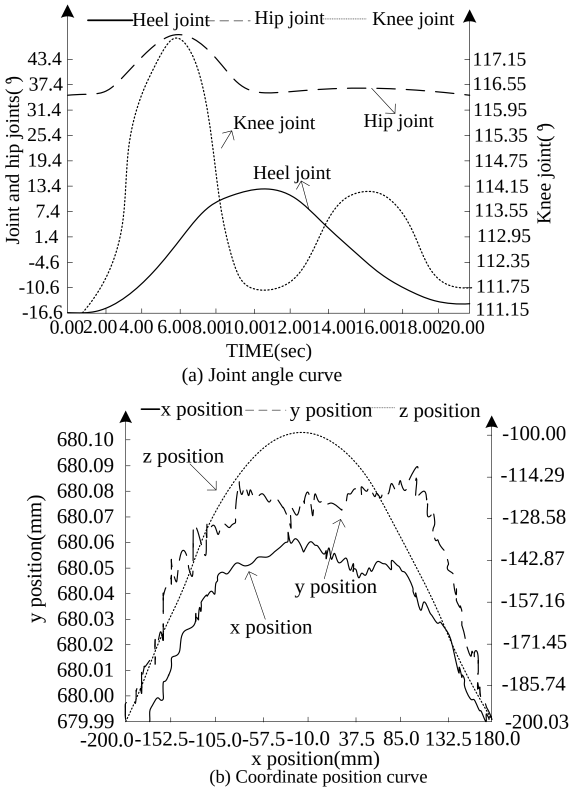
TABLE 1
SYSTEM FUNCTIONAL TEST RESULTS

Use case introduction	Expected results	Actual results	Is it consi stent
The system main interface selects "autonomous navigation mode"	MLHR performs self-help navigation function	MLHR performs self-help navigation function	Yes
The system main interface selects "manual remote control mode"	Enter the second layer of the interface, the progressive state selection interface, which includes three, four, five-legged gait, fixed-point turn gait and free gait	Enter the second layer of the interface, the progressive state selection interface, which includes three, four, five-legged gait, fixed-point turn gait and free gait	Yes
Enter the third level interface and select the robot walking mode	According to the instructions of walking mode, the robot performs "forward", "backward", "left", "right" and "sleep"	According to the instructions of walking mode, the robot performs "forward", "backward", "left", "right" and "sleep"	Yes
Enter the fourth level interface and set the robot's walking velocity	According to the walking velocity instructions, the robot performs "fast", "slow", "acceleration" and "deceleration" walking	According to the walking velocity instructions, the robot performs "fast", "slow", "acceleration" and "deceleration" walking	Yes

C. *Single-legged movement experiment*

Through the movement of each joint, it is determined that the leg movement space is in accordance with the theory, and the joint runs freely. Then in the leg coordinate system, the leg is adjusted to the point with X=-200, Y=680, Z=-200, the intermediate height of 100 mm, then move in the -X direction to point with X = -180, which is similar to the circular motion of the swing support. Through the real-time data acquisition function of UMAC in motion, the angle curve and coordinate position curve of the robot's heel, hip and knee joints under the control can be obtained by the proposed algorithm shown in Figure 7.

According to the data in Figure 7, the algorithm is utilized to control. During the forward swing of the robot leg, the hip and knee joints have a large motion angle, while during the backward motion, the hip and knee joints have a small motion angle. During the whole movement, the movement speed of heel joint did not change significantly. From the position curve analysis, during the movement of the leg in the X direction, the motion error in the Y direction is only 0.1 mm, and the Z direction is highly stable in the support motion. In summary, it can be obtained that the actual motion of the single leg of the robot under the control of the proposed algorithm accords with the forward and inverse kinematics algorithm's solution. The motion process is smooth and stable with high precision.



(a) Joint angle curve

(b) Coordinate position curve

Fig. 7 Curve of joint angle and position coordinate curve

D. Gait walking experiment

To test the effect of the algorithm, the joint servo control system of a MLHR of a certain model is selected as the object to carry out simulation research. The interference input of the system is a small random interference, and the simulation result is shown in Figure 8.

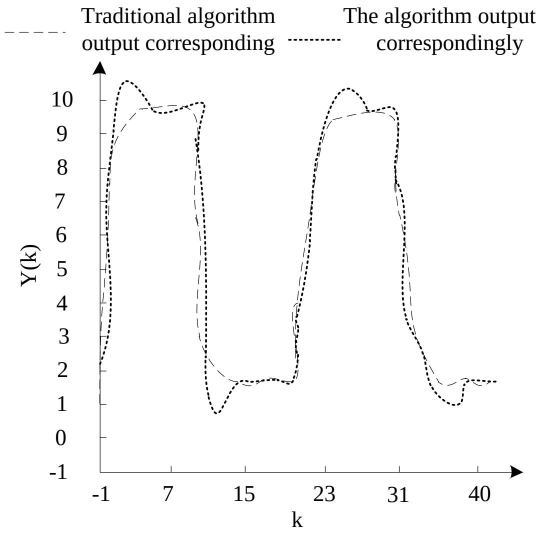


Fig. 8. Experimental results of gait walking for MLHR.

From the results in Figure 8, it is not difficult to see that the robot under the control of the proposed algorithm has no overshoot and fast dynamic response. Thereby, the self-balancing ability of the robot can be improved, and the response coordination of each part thereof can be improved. On the basis of simulation, the experimental research on the simple hierarchical FC of the robot is carried out. The outcomes denote that the algorithm can meet the requirements of dynamic walking. The walking step is 12cm, the walking period is 2.1s, the duty factor is β=0.5, and the walking velocity is about 0.2km/h. Moreover, the robot under the control of the proposed algorithm has good stability, which overcomes the shaking and impact under the control of the traditional algorithm, and the motion is also coordinated.

The experimental condition of the gait rotation of the robot is that the height of the robot is 500mm, the distance between foot end and platform center is 1030mm, and the swing height is 300mm. The triangular rotation gait is used, and each gait cycle is rotated counterclockwise by 10 degrees. Since the robot is hexagonal equilateral distribution, the position curves of the legs in the rotational motion are homologous, and only the second leg's motion position curve is analyzed in Figure 9.

In the Figure 9, the horizontal axis is the X position in the platform coordinate system, the left and the right side are Y and Z positions respectively. The end position of the foot is from the normal standing position (X=0, Y=1080, Z=-400) firstly swings to the position (X=-94, Y=1075.87, Z=-400), then supports to the position (X=-94, Y=1075.87, Z=-400). After that, it swings to the position (X=-94, Y=1075.87, and Z=-400) each time. In the experiment, the hydraulic robots of the multi-legged movements are stable, and there is no sudden change in velocity, which is a good realization of the rotational gait movement.

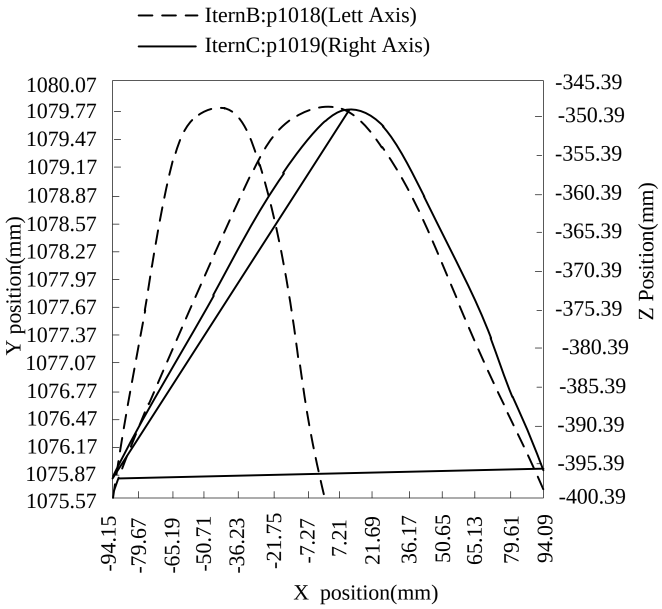


Fig. 9. Position diagram of trigonometric rotation gait.

To prove the function of the raised method on robot gait control (RGC), the references [4], [5] method and the proposed methods are utilized to detect the accuracy of RGC, and the results are shown in Table 2.

TABLE 2
PRECISION OF RGC

Iteration times/times	RGC accuracy/%	Iteration times/times	RGC accuracy/%
100	69.0	87.0	96.0
200	76.5	83.0	98.5
300	83.0	82.0	96.0
400	79.5	79.5	95.0
500	72.4	73.5	96.2

According to table 3, when the number of iterations is 100, the RGC accuracy of references [4], [5] methods and proposed method are 69.0%, 87.0% and 96.0%, respectively; When the number of iterations is 300, the RGC accuracy of references [4], [5] methods and this method are 83.0%, 82.0% and 96.0%, respectively; When the number of iterations is 600, the RGC accuracy of references [4], [5] methods and this method are 69.5%, 79.5% and 98.0%, respectively; The accuracy of RGC of this method is better than that of other methods, which denotes that this method can improve the effect of RGC.

IV. Discussion

In the previous chapter, the obstacle avoidance experiment, functional test experiment, single-legged movement experiment and gait walking experiment of MLHR are carried out. In the obstacle avoidance experiment, the robot can effectively avoid obstacles in different environments; In the functional test experiment, the expected results of the functional test cases are all consistent with the actual test results; In the single-legged movement experiment, there is no sudden change in the velocity of joints motion during exercise, and the motion error is only 0.1mm. In the gait walking experiment, the robot controlled by the proposed algorithm has no overshoot and dynamic response, which overcomes the shaking and impact under the control of traditional algorithms. The action is coordinative, without a sudden change in velocity, and is a good realization of the rotational gait movement. The reason is mainly that the algorithm in this paper carries out the corresponding fuzzy logic reasoning work through the experience and rules, and uses the ambiguous language to describe, which solves the difficulties in building an accurate mathematical model with the increase of system complexity. The algorithm combines with neural network to form closed-loop control, and combines SSM to achieve static and stable gait of MLHR.

Statistical fuzzy logic method, differential evolutionary learning method and the time consumption of FGC of MLHR by this method are displayed in Table 3.

TABLE 3
TIME CONSUMPTION OF FGC FOR MLHR

Iterations/time	Gait control time consumption/s	Iterations/time	Gait control time consumption/s
1000	53.6	56.3	97.2
2000	59.1	59.3	98.1
3000	60.2	60.2	99.2
4000	63.8	61.9	96.3
5000	61.8	63.8	96.5
6000	62.0	58.2	95.0

Table 3 shows that when the amount of iterations is 1000, the FGC time of the MLHR based on fuzzy logic method, differential evolutionary learning method and this method is 53.6s, 56.3s and 97.2s, respectively; When the number of iterations is 5000, the FGC time of the MLHR based on fuzzy logic method, differential evolutionary learning method and this method is 61.8s, 63.8s and 96.5s, respectively; When the number of iterations is 6000, the FGC time of the MLHR based on fuzzy logic method, differential evolutionary learning method and this method is 62.0s, 58.2s and 95.0s, respectively. The above findings express that the proposed method always has high gait control efficiency.

To verify the RGC effect under this method, fuzzy logic method, differential evolution learning method and gait control error of this method are adopted, and the results are displayed in Table 4.

Table 4 shows that when the number of iterations is 1000, the gait control accuracy of the robot under the fuzzy logic method, differential evolutionary learning method and this method can reach 56.2%, 73.8% and 98.2%, respectively; When the number of iterations is 3000, the gait control accuracy of the robot under the fuzzy logic method, differential evolutionary learning method and this method can reach 68.2%, 73.6% and 96.3%, respectively. The method in this paper always has a high gait control accuracy, which shows that the gait control effect of the MLHR in this paper is better.

TABLE 4
GAIT CONTROL ACCURACY OF ROBOT%

Iterations/time	Gait control accuracy of robot%	Iterations/time	Gait control accuracy of robot%
1000	56.2	73.8	98.2
2000	59.7	70.2	90.6
3000	68.2	73.6	96.3
4000	70.1	58.9	98.1
5000	69.3	67.2	95.8
6000	65.6	73.0	99.0

V. Conclusions

The rapid development of science and technology has broadened the space for human activities, and the exploration and research of robots by domestic and foreign researchers has continued to deepen. With the continuous deepening of modern control theory and computer technology research, robots have become a multi-disciplinary and multi-field combination, having an increasingly critical impact on people's daily life. Therefore, a FGC algorithm for MLHRs is put forward in this paper. The experimental and discussion outcomes demonstrate that the robot controlled by this algorithm has no overshoot and dynamic response, overcomes the jitter and impact under the control of traditional algorithms, coordinated motion, no speed mutation, very good The rotating gait movement is realized, which opens up a new path for the FGC of MLHRs. It has certain reference value for the research of MLHRs.

References

[1] H. Yu, H. Gao, L. Ding, et al. "Gait generation with smooth transition using CPG-based locomotion control for hexapod walking robot." *IEEE Transactions on Industrial Electroni*cs, vol. 63, no. 9, pp. 5488-5500, 2016.

[2] Z. Li, S. Xiao, S. S. Ge, et al. "Constrained multilegged robot system modeling and fuzzy control with uncertain kinematics and dynamics incorporating foot force optimization." *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 46, no. 1, pp. 1-15, 2015.

[3] J. Shao, D. Ren, B. Gao, "Recent advances on gait control strategies for hydraulic quadruped robot." *Recent Patents on Mechanical Engineering*, vol. 11, no. 1, pp. 15-23, 2018.

[4] Y. Tan, Y. Irawan, A. Alam, et al. "Buoyancy effect control in multi legged robot locomotion on seabed using integrated impedance-fuzzy logic approach." *Indian Journal of Marine Sciences*, vol. 44, no. 12, pp. 1937-1945, 2015.

[5] B. Aceituno-Cabezas, C. Mastalli, H. Dai, et al. "Simultaneous contact, gait and motion planning for robust multi-legged locomotion via mixed-integer convex optimization." *IEEE Robotics & Automation Letters,* no. 99, PP. 1-1, 2017.

[6] C. F. Juang, Y. H. Chen, Y. H. Jhan, "Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution." *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 611-619. 2015.

[7] J. Pun. 2015. "Application of fuzzy control algorithm in the batching system in stabilized soil mixing." *Automation & Instrumentation,* vol. 25, no. 4, pp. 68-70, 2015.

[8] Y. Zhu, B. Jin, "Compliance control of a legged robot based on improved adaptive control: Method and experiments." *International Journal of Robotics & Automation,* vol. 31, no. 5, pp. 366-373, 2016.

[9] M. Bozic, N. Ducic, G. Djordjevic, "Optimization of when robot running with simulation of neuro-fuzzy control." *International Journal of Simulation Modelling*, vol. 16, no. 1, pp. 19-30, 2017.

[10] S. J. Chang, J. Y. Lee, B. P. Jin, et al. "An online fault tolerant actor-critic neuro-control for a class of nonlinear systems using neural network HJB approach." *International Journal of Control Automation & Systems*, vol. 13, no. 2, pp. 311-318, 2015.

[11] S. Qi, Q. Li, H. T. Wang, et al. "Inertial sensors assisted WiFi indoor positioning method." *Journal of China Academy of Electronics and Information Technology*, vol. 10, no. 1, pp. 102-106, 2015.

[12] D. K. Pratihar, K. Deb, A. Ghosh, "Design of a genetic-fuzzy system for planning crab gaits of a six-legged robot." *Journal of Computing and Information Technology*, vol. 7, no. 1, pp. 7-1, 2000.

[13] A. Irawan, M. M. Alam, Y. Y. Tan, et al. "Center of mass-based admittance control for multi-legged robot walking on the bottom of ocean." *Journal Technology*, vol. 74, no. 9, pp. 1-7, 2015.

[14] Y. M. Lin, H. S. Lin, P. C. Lin, "SLIP-model-based dynamic gait generation in a leg-wheel transformable robot with force control." *IEEE Robotics & Automation Letters*, vol. 2, no. 2, pp. 804-810, 2015.

[15] B. B. Qiu, Z. H. Wang, C. Li, et al. "Fuzzy control strategy for battery equalization charge based on state of charge." *Journal of Power Supply*, vol. 13, no. 2, pp. 113-120, 2015.

[16] Z. H. Zhang, C. Hu, "Multi-model stability control method of underactuated biped robots based on imbalance degrees." *International Journal of Simulation Modelling*, vol. 14, no. 4, pp. 647-657, 2015.

[17] K. Nomura, S. Inagaki, "Cutting a parameter space for a multi-legged robot based on model checking." *SICE Journal of Control Measurement and System Integration*, vol. 10, no. 4, pp. 317-323, 2017.

[18] H. Xie, K. Chen, Y. Yang, et al. "Artificial leg design and control research of a biped robot with heterogeneous legs based on PID control algorithm." *International Journal Bioautomation*, vol. 19, no. 1, pp. 95-106, 2015.

[19] Y. Xu, Z. X. Lin, J. M. Yao, et al. "Target search path fuzzy control of robot navigation." *Computer Simulation*, vol. 33, no. 10, pp. 300-304, 2016.

[20] D. Gong, P. Wang, S. Zhao, et al. "Bionic quadruped robot dynamic gait control strategy based on twenty degrees of freedom." *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 382-388, 2018.

[21] T. Togias, C. Gkournelos, P. Angelakis, M. George, M. Sotiris, "Virtual reality environment for industrial robot control and path design." *Procedia CIRP*, vol. 100, no. 32, pp. 133-138, 2021.

[22] Z. Yan. B. Ouyang, D. Li, et al. "Network intelligence empowered industrial robot control in the f-ran environment." *IEEE Wireless Communications*, vol. 27, no. 2, pp. 58-64, 2020,

[23] Q. Li, L. Zhu, B. Li, et al. "Visual servo control of industrial robot based on convolutional neural network," *Journal of Physics: Conference Series*, vol. 1732, no. 1, pp. 12063-12069, 2021,