sensors

Article

Study of Obstacle-Crossing and Pitch Control Characteristic of a Novel Jumping Robot

Jixue Mo [1,2], Ze Yan [2], Bing Li [1,2], Fengfeng Xi [2] and Yao Li [1,2,*]

[1] State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; 19s053008@stu.hit.edu.cn (J.M.); libing.sgs@hit.edu.cn (B.L.)

[2] School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518052, China; 20s053029@stu.hit.edu.cn (Z.Y.); fengxi@ryerson.ca (F.X.)

[*] Correspondence: liyao2018@hit.edu.cn; Tel.: +86-755-26033485

Abstract: In this study, we demonstrated a novel jumping robot that has the ability of accurate obstacle-crossing jumping and aerial pitch control. The novel robot can quickly leap high into the air with a powerful water jet thruster. The robot was designed to overcome multiple general obstacles via accurate jumping. Then a modified whale optimization algorithm (MWOA) was proposed to determine an optimized jumping trajectory according to the form of obstacles. By comparing with classical intelligent optimization algorithms, the MWOA revealed superiority in convergence rate and precision. Besides, the dynamics model of aerial pitch control was built and its effect was verified by the pitch control experiment. Lastly, the robot's obstacle-crossing experiments were performed and the results validated the robot's good ability of obstacle-crossing and aerial body righting. We believe the optimization of trajectory and the pitch control are of great help for the jumping robot's complex jumping and obstacle-crossing performance.

Keywords: jumping robot; intelligent optimization algorithm; pitch control

check for updates

Citation: Mo, J.; Yan, Z.; Li, B.; Xi, F.; Li, Y. Study of Obstacle-Crossing and Pitch Control Characteristic of a Novel Jumping Robot. *Sensors* **2021**, *21*, 2432. https://doi.org/10.3390/s21072432

Academic Editor: Gabriel Oliver-Codina

Received: 5 February 2021
Accepted: 25 March 2021
Published: 1 April 2021

1. Introduction

In nature, animals use various locomotion methods to travel across different environments, so that their energy consumptions can be effectively minimized [1]. For instance, the frogs usually apply jumping motion to prey or escape from predators, but sometimes they also crawl in a relatively leisurely situation. Thus, the animal's multiple locomotion abilities can motivate the innovative design of structure and locomotion mechanism of a multi-locomotion robot.

The combination of jumping movement and wheeled movement is common in multi-locomotion mode robot, the typical robots include the UMN Scout robot [2], SANDIA wheeled hopping vehicle [3,4], the Boston Dynamics' sand flea robot [5], etc. The jumping movement enables the robot to overcome large-size obstacles and the wheeled movement makes the robot advance rapidly with low energy consumption. Thus, the wheeled jumping robot has better locomotion ability and obstacle-crossing ability compared with other robots.

Inspired by the terrestrial and marine animals, researchers have developed various mobile robots excelling at terrestrial jumping [6–9] or aquatic jumping [10–13]. Most terrestrial jumping robots rely on the surficial reactive force to realize jumping actions while the aquatic jumping robots usually accomplish aquatic jumping in virtue of the surface tension of water or the reactive force from the water jetting process. The gas-powered water jet propulsion scheme proposed in [12] is a proper way to achieve immediate takeoff and effective aquatic jumping on account of its high power density [14]. A well-designed water jet thruster is able to convert the internal high-pressure gas into its own kinetic energy, and the generated thrust is continuous in both terrestrial and aquatic environments since it depends on the surficial reactive force. In addition, the propellant water is environmentally-friendly and easily obtainable. Therefore, the gas-powered water jet propulsion scheme is applicable for terrestrial jumping robots and aquatic jumping robots. The combination of flexible wheels and powerful water jet thrusters on a robot will definitely improve its terrestrial mobility and aerial payload, which is a significative area pending exploration.

Besides wheeled movement and jumping movement, the ability of aerial attitude control (including pitch control and roll control) is also quite important for the jumping robot. The rapid and effective aerial attitude control can increase the jumping robot's possibility of a safe landing. Some small animals, such as lizards [15–17], are good at controlling their body's attitude by swinging tails. Inspired by that, several robot prototypes were designed to research the tail's function in body attitude control. Johnson et al. achieved a legged robot's pitch angle control and safe landing by swaying its tail [18]. Chang-siu also experimented with wheeled robot's tail swinging and pitch control [19]. Kohut et al. researched the tail's effect on dynamic turning characteristics for a miniature legged robot [20].

We have designed a series of water jet thrusters which had a good load capacity and was capable of aquatic jumping and terrestrial jumping [21,22]. However, the water jet thruster can neither achieve aerial attitude control nor accurate trajectory control. Therefore, a novel jumping robot was devised with the proposed water jet thruster in [22], which is the first-ever robotic integration of a wheeled system and a water jet thruster. Moreover, a customized optimization algorithm and aerial pitch control model were proposed to optimize the jumping trajectory and help aerial body righting, respectively. We believe the optimization of trajectory and the pitch control are of great help for the jumping robot's complex jumping and obstacle-crossing performance. Specifically, the decision criteria for overcoming general obstacles was firstly proposed to generate the alternative jumping trajectory set. Then a modified whale optimization method aiming at optimizing the jumping trajectory was put forward, and its superiority was verified by comparing it with other different optimization algorithms. Afterwards, the dynamics model of aerial pitch control was built and its reasonability was verified by the jumping robot's free-falling experiment. Lastly, the jumping robot's jumping and obstacle-crossing experiments were carried out, and the experimental results showed that the proposed robot had the good ability of obstacle-crossing and aerial body righting.

2. Methods

2.1. Structural Design of the Jumping Robot

The prototype of the proposed jumping robot is shown in Figure 1, and its boundary dimension is 487 × 493 × 452 mm^3. It mainly comprises the water jet thruster, shell, front legs, hind legs, assistant legs, battery and control electronics. The water jet thruster is used to accomplish terrestrial and aquatic jumping movement for the jumping robot. The front legs and hind legs were made of high-performance nylon which has good toughness. Then the dampers and foam wheels were installed on the front legs and hind legs, respectively. Hence the robot can achieve compliant landing by buffering of its front legs and hind legs. In addition, the driving wheels of the assistant legs and the driven wheels of the hind legs enables the jumping robot to execute sustainable wheeled movement on general road. Another function of the assistant legs is serving as the aerial attitude control device, which aiming at increasing the jumping robot's possibility of safe landing. It is noteworthy that only the robot's pitch angle can be controlled as the assistant legs have just one rotational degree of freedom. Therefore, the details of the jumping robot's components are listed in Table 1.

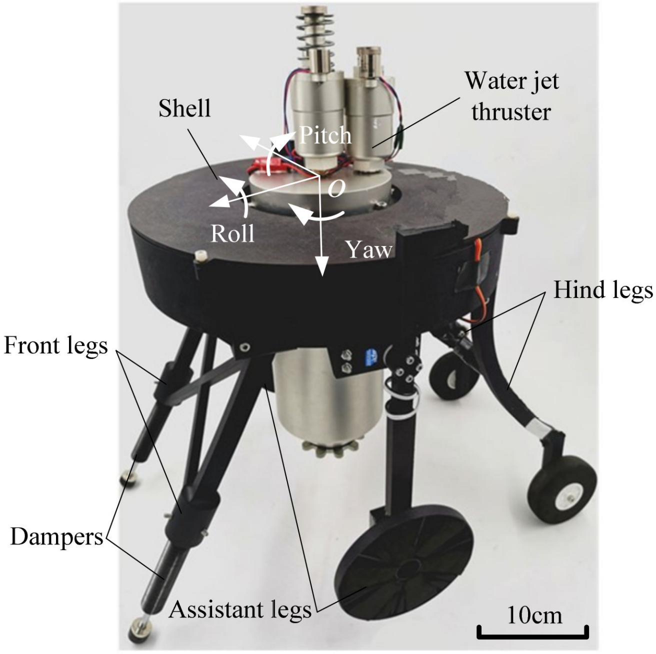

Figure 1. The prototype of the proposed jumping robot.

Table 1. Main components of jumping robot.

Component	Mass (g)	Amount
Water jet thruster	1348	1
Shell	1231	1
Front legs and dampers	485	2
Hind legs, driven wheels and payload	456	2
Servos of Assistant legs	121	2
Assistant legs	182	2
Driving wheels and their servos	256	2
Battery	138	1
Control electronics	147	1
Total	4364	-

The jumping robot's control system consists of the control electronics and upper computer software. It is noteworthy that the microcontroller Arduino UNO R3 was applied as the lower computer, and the upper computer can send commands to the upper computer via wireless communication. The jumping robot used 6 channels of PWM signal on the Arduino UNO R3 to control 2 electromagnets (the electromagnets can activate the robot's jumping action and they are connected with electronic switches which also controlled by PWM signals), 2 servos of assistant legs and 2 servos of driving wheel, respectively. Then the attitude sensor MPU6050 module was used to measure the robot's accelerations and attitude angles on 3 axes. In addition, the Bluetooth module was applied to achieve the communication between Arduino and upper computer. Lastly, the working process of the jumping robot's control system are shown in Figure 2.

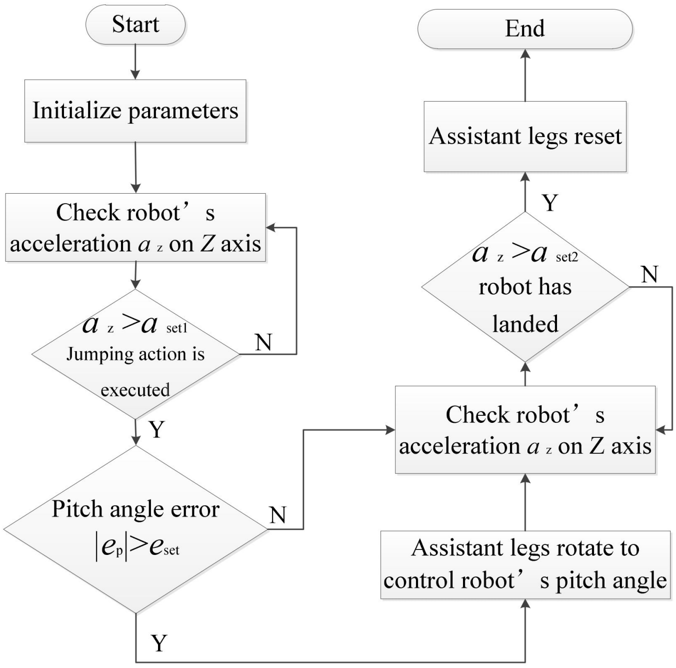
Start
Initialize parameters
Check robot' s acceleration a z on Z axis
a z >a set1 Jumping action is executed
N
Y
Pitch angle error |ep|>eset
N
Y
End
Assistant legs reset
Y
a z >a set2 robot has landed
N
Check robot' s acceleration a z on Z axis
Assistant legs rotate to control robot' s pitch angle

Figure 2. The working process of jumping robot's control system. a_{set1} and a_{set2} are two set values of acceleration. e_{p} and e_{set} are the measured value and set value of pitch angle error, respectively, and 0° is the expected value of pitch angle.

2.2. Decision Criteria of Overcoming General Obstacles

When the jumping robot jumps in the air, the pressure energy of high-pressure gas is converted into kinetic energy of the robot and the propellant water [22]. Supposing the bottle plug of the water jet thruster can be released anytime during the pressurization process, and the duration of the water jetting process is quite short, then the maximum pressure of pressurization $P_{\max}$ determines the take-off velocity v_0, and the jumping trajectory can be treated as parabola.

As shown in Figure 3, taking the robot's gravity center o as the original point and the central axis of water jet thruster as the y axis, the carrier coordinate system xoy can be established; taking the initial location of robot's gravity center O as the original point and the vertical direction as Y axis, the ground coordinate system XOY can be established. Then there are two controllable parameters while jumping: the take-off angle φ_0 and the take-off velocity v_0, which are shown in Figure 3a.

It is obvious that different values of φ_0 and v_0 can generate different parabolic trajectory. Supposing the parabolic peak's coordinate in ground coordinate system is $(x_{\text{p}}, y_{\text{p}})$, then it can be obtained that:

$$\begin{cases} x_{\text{p}} = v_0 \cos \varphi_0 \sqrt{2y_{\text{p}}/g} \\ y_{\text{p}} = (v_0 \sin \varphi_0)^2/2g \end{cases}. \tag{1}$$

As the parabola goes through original point, the parabolic equation can be written as:

$$y = -\frac{y_{\text{p}}}{x_{\text{p}}^2}(x - x_{\text{p}})^2 + y_{\text{p}}. \tag{2}$$

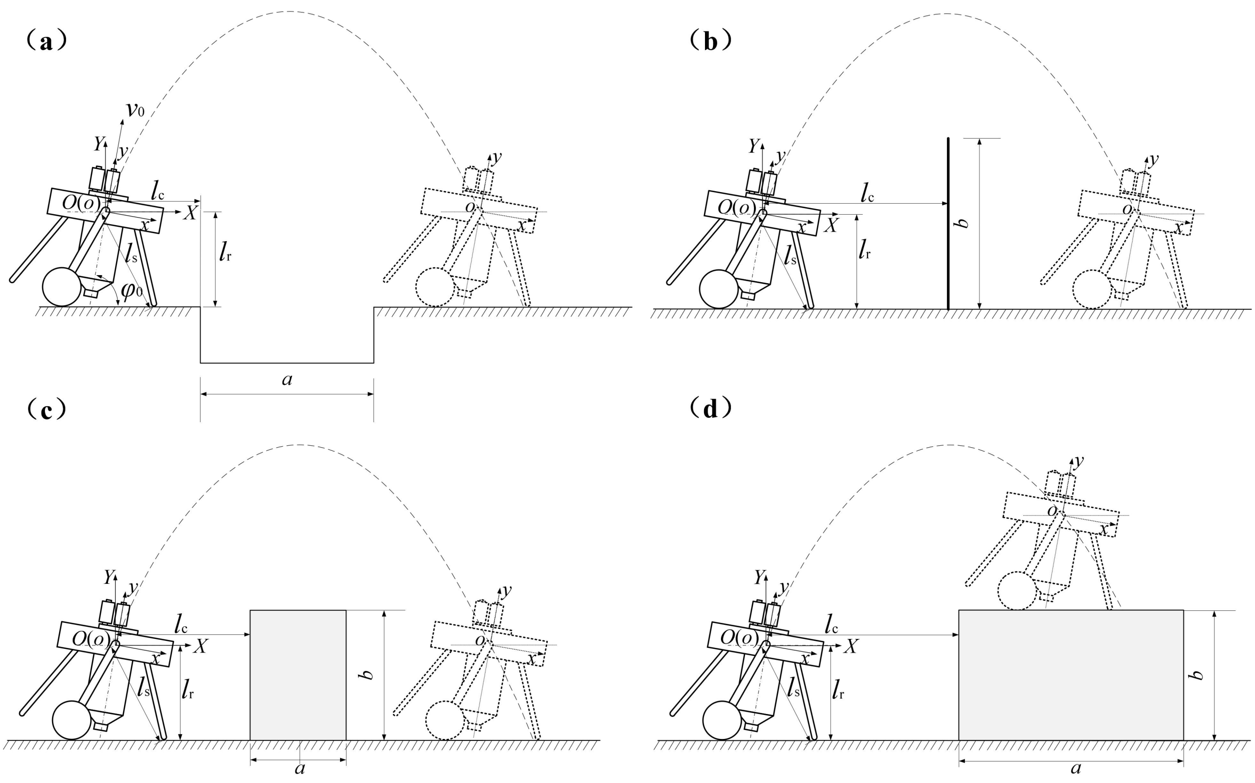

Figure 3. The schematic diagram of jumping robot's overcoming different obstacles. (**a**) Jumping robot jumps over ditch. (**b**) Jumping robot jumps over wall. (**c**) Jumping robot jumps over platform. (**d**) Jumping robot jumps onto platform.

In actual situation, the jumping robot may mainly come across three types of obstacles: ditch, wall and platform. Therefore, there are four types of obstacle-crossing tasks: jumping over ditch, jumping over wall, jumping over platform and jumping onto platform (Figure 3). It is noteworthy that the ditch and wall are special cases of platform, as the ditch and wall can be treated as platforms with little height and width respectively. Therefore, the discussions below are for overcoming the platform.

Supposing the platform's width and height as a and b respectively, the distance between its left side and the origin O is l_c, the height of origin O from ground is l_r. In consideration of the safety while landing, there should be a safety distance l_s (Figure 3c). Therefore, the necessary conditions of jumping over the platform are: the coordinates of the platform's top vertexes should be located below the parabola; the minimum distance between the platform's top vertexes and the parabola should be no more than the safety distance l_s. Thus, the decision criteria can be written as:

$$\begin{cases} -\frac{y_p}{x_p^2}(l_c - x_p)^2 + y_p > b - l_r \\ -\frac{y_p}{x_p^2}(l_c + a - x_p)^2 + y_p > b - l_r \\ \min(\sqrt{(x - l_c)^2 + (y - b + l_r)^2}) > l_s \\ \min(\sqrt{(x - l_c - a)^2 + (y - b + l_r)^2}) > l_s \end{cases}. \tag{3}$$

Then the constraint conditions are:

$$\begin{cases} y = -\frac{y_p}{x_p^2}(x - x_p)^2 + y_p \\ 0 \le x \le \max(2x_p, l_c + l_s + a) \end{cases}. \tag{4}$$

In the case of jumping onto the platform, the necessary condition is relatively simple: the coordinates of the platform's top left vertex should be located below the parabola; the minimum distance between the platform's top left vertex and the parabola should be no more than the safety distance l_s. Thus, the decision criteria can be written as:

$$\begin{cases} -\frac{y_p}{x_p^2}(l_c - x_p)^2 + y_p > b - l_r \\ \min(\sqrt{(x - l_c)^2 + (y - b + l_r)^2}) > l_s \end{cases}. \tag{5}$$

Then the constraint conditions are:

$$\begin{cases} y = -\frac{y_p}{x_p^2}(x - x_p)^2 + y_p \\ 0 \le x \le \max(x_p + \sqrt{x_p^2(y_p - b)/y_p}, l_s + l_c) \end{cases}. \tag{6}$$

Hence the value ranges of φ_0 and v_0 can be determined according to Equations (3)–(6) in the case of jumping over the platform and jumping onto the platform.

2.3. Modified Whale Optimization Algorithm

After determining the value range of the jumping robot's φ_0 and v_0, there will be an obstacle-crossing trajectory set for a particular obstacle. It is necessary to optimize the jumping trajectory according to certain indicators by proper optimization algorithms. The whale optimization algorithm (WOA) was proposed by Mirjalili and Lewis in 2016 [23]. It has good performance in terms of convergence rate and convergence precision. However, it also has the problem of outputting a local optimal solution when dealing with complex problems. Thus, a modified whale optimization algorithm was proposed and applied in the optimization of the jumping trajectory.

2.3.1. Standard Whale Optimization Algorithm

The whale optimization algorithm simulates the foraging behavior of humpback whales. The algorithm can be divided into three stages: encircling prey stage, bubbling-net attacking stage and searching for prey.

1. Encircling prey stage. Supposing the scale of the whale population is N, the dimension of the searching space is d, then the whale individual i can be expressed as $X_i = (x_i{}^1, x_i{}^2, \ldots, x_i{}^d)$, $i = 1, 2, \ldots, N$. The prey's position corresponds to the global optimal solution of objective function. Therefore, the mathematic model of the encircling prey stage can be expressed as:

$$\boldsymbol{D} = |\boldsymbol{C} \cdot \boldsymbol{X}_p(t) - \boldsymbol{X}(t)|, \tag{7}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_p(t) - \boldsymbol{A} \cdot \boldsymbol{D}, \tag{8}$$

where $\boldsymbol{D}$ is the distance vector between the current optimal solution and the searching individual, t is the current iterations, $\boldsymbol{X}$ (t) is the position vector of whale individual and $\boldsymbol{X}_p(t)$ is the current optimal solution. The coefficient vector $\boldsymbol{A}$ and $\boldsymbol{C}$ can be calculated by

$$\boldsymbol{A} = 2a \cdot \boldsymbol{r}_1 - a, \tag{9}$$

$$\boldsymbol{C} = 2 \cdot \boldsymbol{r}_2, \tag{10}$$

where $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are random vectors in [0,1], and a linearly decreases from 2 to 0 with t's growth:

$$a(t) = 2(1 - \frac{t}{t_{\max}}). \tag{11}$$

2. Bubbling-net attacking stage. This stage stimulates the process that the whale individual moves spirally around the current prey to approach it. The mathematical model of the spiral motion is:

$$\boldsymbol{X}(t+1) = \boldsymbol{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \boldsymbol{X}_p(t) \; (p > 0.5), \tag{12}$$

where $\boldsymbol{D'}$ = $|\boldsymbol{X}_{\mathrm{p}}(t) - \boldsymbol{X}(t)|$ is the distance vector between the current optimal solution and the searching individual, b is a constant of defining the shape of the logarithmic spiral and l is a random number in [−1,1]. In order to synchronize the behaviors of encircling prey and bubbling-net attack, a probability variable p in [0,1] is introduced, thus the mathematical model can be written as:

$$\boldsymbol{X}(t+1) = \begin{cases} \boldsymbol{X}_p(t) - \boldsymbol{A}\cdot\boldsymbol{D}\ (p < 0.5) \\ \boldsymbol{D'}\cdot e^{bl}\cdot\cos(2\pi l) + X_p(t)\ (p > 0.5) \end{cases}. \tag{13}$$

3. Searching for prey stage. The whale individual decides encircling for prey or searching for prey by the value of $|\boldsymbol{A}|$. When $|\boldsymbol{A}| > 1$, the whale cannot clearly determine the prey's position. Therefore, it is necessary to pick a random individual in place of the prey, that is:

$$\boldsymbol{D''} = |\boldsymbol{C}\cdot\boldsymbol{X}_{rand} - \boldsymbol{X}|, \tag{14}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{rand} - \boldsymbol{A}\cdot\boldsymbol{D''}. \tag{15}$$

2.3.2. Improvement Strategy

1. Tent mapping strategy. The standard WOA generates an initial population by random method, this may lead to its poor diversity. Therefore, a new chaotic mapping method named tent mapping was applied [24]. For a population with scale N and dimension d, the chaotic sequence $y = \{y_d, d = 1, 2, \ldots, d\}$, $y_d = \{y_{id}, i = 1, 2, \ldots, N\}$ can be firstly obtained by tent mapping with the equation

$$y_{i+1,d} = \begin{cases} 2y_{id},\ y_{id} < 0.5 \\ 2(1-y_{id}),\ y_{id} \geq 0.5 \end{cases}. \tag{16}$$

Mapping the chaotic sequence to solution space, a population $X = \{X_i, i = 1, 2, \ldots, N\}$, $X_i = \{X_{id}, d = 1, 2, \ldots, d\}$ can be obtained, and

$$X_{id} = X_{\min,d} + y_{id}(X_{\max,d} - X_{\min,d}), \tag{17}$$

where $X_{max,d}$ and $X_{min,d}$ are the upper and lower limit of X_{id} respectively. Then the reverse population OX can be obtained according to X by

$$OX_{id} = X_{\max,d} + X_{\min,d} - X_{id}. \tag{18}$$

Lastly, merge the population X and OX, and sort the new population according to the objective function's fitness, the top N individuals can be taken as the initial population.

2. Non-linear convergence factor and self-adaptive weight strategy. The linear decrease of convergence factor a in standard WOA cannot coordinate global searching and local exploitation. Therefore, it is better to make the descending speed of a decrease with the progress of iterations t. The proposed equation of non-linear convergence factor is

$$a = 2[1 - \cos(\mu\frac{t}{t_{\max}}\pi + \theta)], \tag{19}$$

where μ = 0.5 and θ =−π/2 are relevant parameters. Besides, the standard WOA does not take the prey' guiding force diversity for the whale's position update with the progress of iterations into consideration. Therefore, a self-adaptive weight can be introduced in the position update equation to utilize the current optimal solution effectively. The equation of self-adaptive weight is:

$$\omega(t) = 1 - \frac{e^{t/t_{\max}} - 1}{e - 1}. \tag{20}$$

Then Equations (8) and (12) can be modified as follows:

$$\boldsymbol{X}(t+1) = \omega(t)\cdot\boldsymbol{X}_{\mathrm{p}}(t) - \boldsymbol{A}\cdot\boldsymbol{D}, \tag{21}$$

$$\boldsymbol{X}(t+1) = \boldsymbol{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \omega(t) \cdot \boldsymbol{X}_p(t)\ (p > 0.5). \tag{22}$$

3. Diversity variation strategy. The population's diversity will gradually decrease with the iterative process in standard WOA, which may lead to the algorithm's premature convergence. Therefore, the variation operation for the population should be conducted when the diversity is below the threshold value. Supposing the fitness value of individual I is f_i, $f_{\max}$, $f_{\min}$ and f_{mean} are the maximum, minimum and mean fitness values of the current population respectively. Then the variance of current population is:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{n} \left(\frac{f_i - f_{\text{mean}}}{f_{\max} - f_{\min}} \right)^2. \tag{23}$$

When the variance is below threshold value, the variation operation can be conducted as follows:

$$\boldsymbol{X}(t+1) = r_3 \cdot \left[\boldsymbol{X}_p(t) - \boldsymbol{X}(t) \right] + r_4 \cdot \left[\boldsymbol{X}'(t) - \boldsymbol{X}(t) \right], \tag{24}$$

where r_3 and r_4 are random numbers in [0,1], and $\boldsymbol{X}'(t)$ is a random individual in the population.

Therefore, the flow diagram of MWOA is presented in Figure 4.

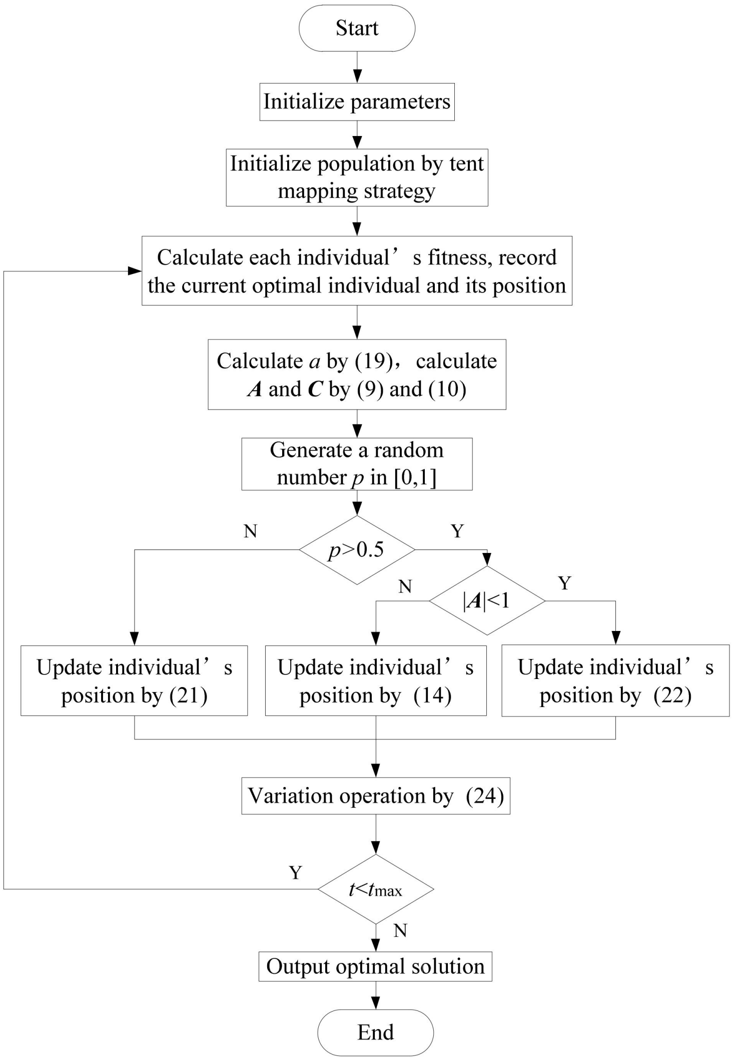
Start
Initialize parameters
Initialize population by tent mapping strategy
Calculate each individual’ s fitness, record the current optimal individual and its position
Calculate a by (19)，calculate $\boldsymbol{A}$ and $\boldsymbol{C}$ by (9) and (10)
Generate a random number p in [0,1]
N
p>0.5
Y
N
|$\boldsymbol{A}$|<1
Y
Update individual’ s position by (21)
Update individual’ s position by (14)
Update individual’ s position by (22)
Variation operation by (24)
Y
t<$t_{\max}$
N
Output optimal solution
End

Figure 4. Flow diagram of modified whale optimization algorithm.

2.3.3. Performance Test

In order to test the optimization performance of MWOA, 10 benchmark functions were chosen for the performance test. The details of the benchmark functions are listed in Table A1 (Appendix A). It is noteworthy that f_1~f_4 are unimodal functions, f_5~f_8 are multimodal functions and f_9~f_{10} are fixed dimension functions, respectively. The performance test was conducted in two aspects: 1. Test the performance improvement effect of each improvement strategy. 2. Compare the convergence rate and precision of MWOA with standard WOA and another two intelligent optimization algorithms, including DA (Dragonfly Algorithm) [25] and HHO (Harris Hawk Optimization) [26]. The population scale and maximum iterations in both performance tests were 30 and 500 respectively; each benchmark function was tested 30 times, and then the mean value and standard deviation were calculated.

2.3.4. Fitness Function

After determining the specific optimization algorithm, the conditions and objective of jumping trajectory optimization should also be determined. Thus, the condition was: the details of jumping robot's location, obstacle's location and size were known; the objective was: to optimize the two take-off parameters (take-off angle φ_0 and take-off velocity v_0) under the fitness function for a specific obstacle-crossing task.

1. Constraint condition of take-off parameters. The jumping trajectory is determined by the take-off parameters φ_0 (rad) and v_0 (m/s), while v_0 was determined by the pressurization pressure, jumping robot's total mass and the mass of propellant water inside the water jet thruster. The mapping relationship between v_0 and these variables can be obtained by the theoretical jumping model in [22]. Then it is noticeable that the water jet thruster's propulsive performance would not be so good if φ_0 takes a small value, because the gas–water interface inside the thruster would have a relatively small angle with the thruster's inner wall and the effect of the water-jetting process would be weaken. Thus, according to the actual situation, the constraint condition of φ_0 and v_0 are given to be:

$$\begin{cases} \pi/3 < \varphi_0 < \pi/2 \\ 4 < v_0 < 8 \end{cases}. \tag{25}$$

2. Fitness function. The fitness function comprises three parts: horizontal jumping distance, vertical jumping distance and landing safety. If the robot confronts a ditch or a flat obstacle, the horizontal jumping distance is its priority. Thus, the fitness function of the horizontal jumping ability is

$$f_{o1} = \frac{2v_0^2 \cos\varphi_0 sin\varphi_0/g}{\max(2v_0^2 \cos\varphi_0 sin\varphi_0/g)}, \tag{26}$$

where the denominator is the maximum value of the numerator in the constraint condition. If the robot confronts a wall or a slim obstacle, the vertical jumping distance is its priority. Thus, the fitness function of the vertical jumping ability is

$$f_{o2} = \frac{v_0^2 sin^2\varphi_0}{v_{0\max}^2}, \tag{27}$$

where $v_{0\max}$ is the upper limit of v_0. If the robot lands on hard ground with high speed, it can be impacted violently and toppled easily. Thus, the fitness function of landing safety is

$$f_{o3} = 1 - \frac{(v_0 - 4)^2}{32} - \frac{(\varphi_0 - 0.45\pi)^2}{2(0.1167\pi)^2}. \tag{28}$$

It is noteworthy that these three indicators restrict each other, and they cannot achieve the best value simultaneously. Therefore, each indicator can be weighted and the total fitness function is:

$$F(\varphi_0, v_0) = \sum_{i=1}^{3} \omega_i f_{oi}(\varphi_0, v_0), \tag{29}$$

where ω_i is the weight coefficient and $\omega_1 + \omega_2 + \omega_3 = 1$.

2.4. Dynamics Model of Aerial Pitch Control

The dynamics model for the jumping robot's aerial pitch control belongs to the dynamics of coupled rigid bodies while free falling, which were studied by relevant researchers. Yang et al. built the dynamics model of two rigid bodies connected by a spherical joint and then proposed the controllers by input–output linearization [27]. Agrawal et al. applied differential flatness to achieve the control of two coupled rigid bodies [28]. Chang-Siu et al. put forward the non-linear control model for a robot with a two-link tail and two degree-of-freedom actuations [29].

In order to achieve a reliable and effective aerial pitch control for the jumping robot, the dynamics model was first built. For simplification, the robot was seen as being composed of two parts: the main body and the assistant legs; then the robot only rotates in the pitching direction. Since the water jet thruster's propellant water is drained in a short period of time, the dynamics model assumes that the robotic system does not contain propellant water during the aerial pitch control process. As shown in Figure 5, the assistant leg's centroid is A, its center of rotation on the main body is C, the main body's centroid is B and the whole robot's centroid is O'. Then a dynamic coordinate system $O'X'Y'$ was built where the X' axis and Y' axis are along the horizontal direction and the vertical direction, respectively.

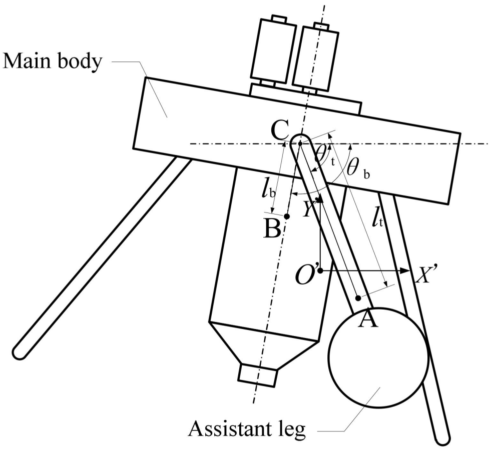

Figure 5. The schematic diagram of jumping robot in the air.

Then the robot system's dynamics model can be derived by the Euler–Lagrange method. Due to the establishment of $O'X'Y'$, the robot's translational movement can be decoupled from its rotational movement [30]. As the robot's translational movement is a parabolic movement [31], only the rotational movement is discussed in this section. Therefore, the Lagrangian of the robot system can be written as [32]:

$$\begin{cases} L = \frac{1}{2}\left(m_t {v_A}^2 + I_t \dot{\theta}_t^2\right) + \frac{1}{2}\left(m_b {v_B}^2 + I_b \dot{\theta}_b^2\right) \\ \quad = \frac{1}{2}\left(m_t {v_A}^2 + m_b {v_B}^2 + I_t \dot{\theta}_t^2 + I_b \dot{\theta}_b^2\right) \\ v_A = \frac{m_b}{2m_t + m_b}\left(\dot{\theta}_t l_t - \dot{\theta}_b l_b\right) \\ v_B = \frac{m_t}{2m_t + m_b}\left(\dot{\theta}_t l_t - \dot{\theta}_b l_b\right) \end{cases}, \tag{30}$$

where m_{b} and m_{t} are the mass of main body and assistant legs, respectively; I_{b} and I_{t} are the rotational inertia of main body and the assistant legs, respectively; θ_{b} and θ_{t} are the angle of main body and assistant leg relative to X' axis, respectively, and $\theta_{\mathrm{m}} = \theta_{\mathrm{b}} - \theta_{\mathrm{t}}$ is the angle of main body relative to assistant leg.

Then applying the Euler–Lagrange method to the robot system, it can be obtained that:

$$\begin{cases} \dfrac{\mathrm{d}\left(\frac{\partial L}{\partial \dot{\theta}_{\mathrm{t}}}\right)}{\mathrm{d}t} - \dfrac{\partial L}{\partial \theta_{\mathrm{t}}} = \tau \\ \dfrac{\mathrm{d}\left(\frac{\partial L}{\partial \dot{\theta}_{\mathrm{b}}}\right)}{\mathrm{d}t} - \dfrac{\partial L}{\partial \theta_{\mathrm{b}}} = -\tau \end{cases}. \tag{31}$$

It is noteworthy that there is only one external input (the torque τ of assistant leg's servo), so the dynamics Equations can be obtained by substituting (30) into (31):

$$\begin{cases} M\ddot{\theta}_{\mathrm{t}} - K\cos\theta_{\mathrm{m}}\ddot{\theta}_{\mathrm{b}} + K\sin\theta_{\mathrm{m}}\dot{\theta}_{\mathrm{b}}^{2} = \tau \\ N\ddot{\theta}_{\mathrm{b}} - K\cos\theta_{\mathrm{m}}\ddot{\theta}_{\mathrm{t}} - K\sin\theta_{\mathrm{m}}\dot{\theta}_{\mathrm{t}}^{2} = -\tau \\ M = 2I_{\mathrm{t}} + 2m_{\mathrm{t}}m_{\mathrm{b}}/(m_{\mathrm{t}} + m_{\mathrm{b}})l_{\mathrm{t}}{}^{2} \\ N = I_{\mathrm{b}} + 2m_{\mathrm{t}}m_{\mathrm{b}}/(m_{\mathrm{t}} + m_{\mathrm{b}})l_{\mathrm{b}}{}^{2} \\ K = 2m_{\mathrm{t}}m_{\mathrm{b}}/(m_{\mathrm{t}} + m_{\mathrm{b}})l_{\mathrm{b}}l_{\mathrm{t}} \end{cases}. \tag{32}$$

For that dynamics equation, there are two controllable parameters, θ_{b} and θ_{t}, while there is only one input parameter, τ. However, only θ_{b} should be precisely controlled, so the two dynamics equations in (32) can be properly transformed to one which only contains the parameter $\theta_{\mathrm{m}} = \theta_{\mathrm{b}} - \theta_{\mathrm{t}}$. Firstly, the equations of θ_{b} and θ_{t} can be attained according to (32) as

$$\begin{cases} \ddot{\theta}_{\mathrm{b}} = \frac{SM\dot{\theta}^{2}{}_{\mathrm{t}} - Q\tau - SK\cos\theta_{\mathrm{m}}\dot{\theta}_{\mathrm{b}}^{2}}{P} \\ \ddot{\theta}_{\mathrm{t}} = \frac{-SN\dot{\theta}^{2}{}_{\mathrm{b}} + R\tau + SK\cos\theta_{\mathrm{m}}\dot{\theta}_{\mathrm{t}}^{2}}{P} \\ P = MN - K^{2}\cos^{2}\theta_{\mathrm{m}} \\ S = K\sin\theta_{\mathrm{m}} \\ Q = M - K\cos\theta_{\mathrm{m}} \\ R = N - K\cos\theta_{\mathrm{m}} \end{cases}. \tag{33}$$

As $\ddot{\theta}_{\mathrm{m}} = \ddot{\theta}_{\mathrm{b}} - \ddot{\theta}_{\mathrm{t}}$, thus it can be derived from (33) that

$$\ddot{\theta}_{\mathrm{m}} = \ddot{\theta}_{\mathrm{b}} - \ddot{\theta}_{\mathrm{t}} = \frac{SQ\dot{\theta}^{2}{}_{\mathrm{t}} + SR\dot{\theta}^{2}{}_{\mathrm{b}} - Q\tau - R\tau}{P}. \tag{34}$$

Then the parameters $\dot{\theta}_{\mathrm{b}}$ and $\dot{\theta}_{\mathrm{t}}$ can be eliminated by the theorem of angular momentum conservation. Supposing the robotic system's total angular momentum is zero, then the robotic system's angular momentum is

$$H_0 = \frac{\partial L}{\partial \dot{\theta}_{\mathrm{t}}} + \frac{\partial L}{\partial \dot{\theta}_{\mathrm{b}}} = (M - Kcos\theta_{\mathrm{m}})\dot{\theta}_{\mathrm{t}} + (N - Kcos\theta_{\mathrm{m}})\dot{\theta}_{\mathrm{b}} = 0. \tag{35}$$

Thus, the equations of $\dot{\theta}_{\mathrm{b}}$ and $\dot{\theta}_{\mathrm{t}}$ can be derived as

$$\begin{cases} \dot{\theta}_{\mathrm{b}} = \frac{Q}{R+Q}\dot{\theta}_{\mathrm{m}} \\ \dot{\theta}_{\mathrm{t}} = \frac{-R}{R+Q}\dot{\theta}_{\mathrm{m}} \end{cases}. \tag{36}$$

Taking the integral of the first Equation of (36), it can be obtained that

$$\theta_{\mathrm{b}}(t) = \theta_{\mathrm{b}}(0) + \int_0^t \frac{Q}{(R+Q)} \dot{\theta}_{\mathrm{m}} \mathrm{d}t, \tag{37}$$

where $\theta_{\mathrm{b}}(0)$ is the initial value of θ_{b} of the aerial pitch control process. Then replacing the integration variable t with θ_{m}, it can be obtained that:

$$\theta_{\mathrm{b}}(t) = \int_{\theta_{\mathrm{m0}}}^{\theta_{\mathrm{mt}}} \frac{Q}{(R+Q)} \mathrm{d}\theta_{\mathrm{m}} + \theta_{\mathrm{b}}(0), \tag{38}$$

where θ_{m0} and θ_{mt} are θ_{m}'s values at the initial and end moment of the aerial pitch control process respectively. Then combining Equations (32) and (33), the antiderivative of $Q/(R + Q)$ in (38) can be calculated by MATLAB 2016 [33], and the calculation result is

$$\begin{cases} f(\theta_{\mathrm{m}}) = \frac{\theta_{\mathrm{m}}}{2} - \frac{U \cdot V}{W} \\ U = \ln[\frac{\cos\theta_{\mathrm{m}}(M+N) - 2K + \sin\theta_{\mathrm{m}}\sqrt{4K^2 - (M+N)^2}}{M+N-2K\cos\theta_{\mathrm{m}}}] \\ V = K(M+N) - 2KM \\ W = 2K\sqrt{4K^2 - (M+N)^2} \end{cases}, \tag{39}$$

Thus, Equation (38) can be transformed to be

$$\theta_{\mathrm{b}}(t) - \theta_{\mathrm{b}}(0) = f(\theta_{\mathrm{mt}}) - f(\theta_{\mathrm{m0}}). \tag{40}$$

Therefore, once the initial value $\theta_{\mathrm{b}}(0)$, final value $\theta_{\mathrm{b}}(t)$ of θ_{b} and the initial value θ_{m0} of θ_{m} are determined, the final value $\theta_{\mathrm{m}t}$ of θ_{m} can be calculated and then the necessary rotation angle of assistant leg $\Delta\theta_{\mathrm{t}} = \theta_{\mathrm{t}}(t) - \theta_{\mathrm{t}}(0)$ can also be determined. In most cases, $\theta_{\mathrm{b}}(t) = -\pi/2$, and the ranges of $\theta_{\mathrm{b}}(0)$ and θ_{m0}, were given to be $[-\pi,0]$ and $[-\pi/2, \pi/2]$, respectively.

2.5. Aerial Pitch Control Experiments of Jumping Robot

Since the dynamics model of aerial pitch control of the jumping robot were proposed, its correctness and reliability should be verified by experiments. The experimental method lets the robot fall freely from the given initial height and records the pitch angle's variation with or without the effect of aerial pitch control movement. The experiments were divided into two groups and each group contained five experiments. There was 580 mL propellant water in the water jet thruster's inner tank in the Group 2, while there was no water in Group 1. Thus, the experimental scheme of aerial pitch control was shown in Table 2. It is noteworthy that there were no aerial pitch control actions in experiment No. 5 and No. 10, which were meant to be the control experiments of experiment No. 4 and No. 9, respectively; besides, the robot's pitch angle $\theta_{\mathrm{p}} = \theta_{\mathrm{b}} + \pi/2$. In order to prevent obvious error in the robot's roll direction during the free falling process, the robot's initial roll angle was controlled to be zero and its center of gravity in the roll direction was also balanced.

Table 2. The experimental scheme of aerial pitch control of jumping robot.

	Experiment No.	Pitch Angle (°)	Initial Falling Height (m)	Assistant Leg's Theoretical Rotation Angle (°)
Group 1	1	−12	1.35	−34.2
	2	−12	1.4	−34.2
	3	−17	1.35	−48.5
	4	−17	1.4	−48.5
	5	−17	1.4	/
Group 2	6	−15	1.05	−49.3
	7	−15	1.3	−49.3
	8	15	1.05	49.3
	9	15	1.3	49.3
	10	15	1.3	/

The assistant leg's theoretical rotation angle of each experiment can be calculated from Equation (40). For the jumping robot, the related parameters in (40) are: m_t = 0.438 kg, m_{b_1} = 3.922 kg, m_{b_2} = 4.502 kg, l_t = 0.146 m, l_b = 0.041 m, I_t = 0.00668 kg·m^2, I_{b_1} = 0.085 kg·m^2 and I_{b_2} = 0.087 kg·m^2. It is noteworthy that m_{b_1} and m_{b_2} are the main body's mass in Group 1 and Group 2; I_t are the total rotational inertia of the robot's assistant legs as they were meant to rotate simultaneously; I_{b_1} and I_{b_2} are the main body's rotational inertia in Group 1 and Group 2. The expected value of the main body and the initial angle of the main body relative to the assistant leg were given to be $\theta_b(t)= -\pi/2$ and $\theta_{m0} = 0$, respectively. Hence for each experiment in Group 1 and Group 2, the corresponding assistant leg's theoretical rotation angle $\Delta\theta_t = \theta_t(t) - \theta_t(0)$ can be calculated according to Equation (40), and the calculation results were listed in Table 2.

In addition, the falling and aerial pitch control process of the jumping robot was recorded by a high-speed camera (i-speed 221, iX Cameras, Britain), and the shooting frame rate was 400 Hz. The variation of the robot's pitch angle θ_p was obtained from the analysis of experimental video in the motion analysis software ProAnalyst, and the sampling frequency was 400 Hz; the robot's height from the ground was measured by a telemeter rod.

2.6. Jumping Experiments of the Robot

2.6.1. Trajectory Accuracy Verification Experiments

In order to ensure the experimental jumping trajectory's accuracy, it is necessary to carry out the trajectory accuracy verification experiments, which contained four terrestrial jumping experiments of the jumping robot with different take-off parameters (take-off angle φ_0 and take-off velocity v_0). It is noteworthy that the terrestrial jumping movements were conducted by the water jetting process of the water jet thruster's inner tank. The take-off velocity v_0 was positively related to the maximum pressurization pressure P_{max} when the robot's total mass and propellant water's mass were determined, and the mapping relationship between v_0 and P_{max} were given by the theoretical jumping model in [22]. Thus, the take-off parameters of four experiments were: a. φ_0 = 87°, P_{max} = 1.81 MPa; b. φ_0 = 87°, P_{max} = 2.23 MPa; c. φ_0 = 78°, P_{max} = 1.81 MPa; d. φ_0 = 78°, P_{max} = 2.23 MPa.

In addition, the volumes of propellant water in four experiments were all 580 mL, and the jumping robot's total mass was 4.36 kg. Thus, the corresponding theoretical take-off velocities with given maximum pressurization pressures of 1.81 MPa and 2.23 MPa can be calculated to be 5.36 m/s and 6.46 m/s, respectively.

In the robot's terrestrial jumping experiment, the propellant water was firstly injected into the water jet thruster. Then the robot's initial attitude was adjusted according to the given take-off angle. The robot started jumping movement a while after the liquid nitrogen was injected and the water jet thruster's inner tank was sealed. The jumping process was recorded by a digital camera (FDR-AX45, Sony, Tokyo) and the experimental jumping trajectory of robot's centroid was extracted from the experimental video by ProAnalyst.

2.6.2. Obstacle-Crossing Experiments

After the accuracy of robot's experimental jumping trajectory was verified, the obstacle-crossing experiments can be conducted with the third optimized take-off parameters whose priority was landing safety. These experiments were meant to verify the robot's abilities of obstacle-crossing and aerial pitch control. Similarly, the volume of propellant water inside the water jet thruster's inner tank was 580 mL, and the jumping robot's total mass was 4.36 kg. As the optimized take-off parameters of the experiments of jumping onto the platform and jumping over the platform were both φ_0 = 76.0° and v_0 = 5.57 m/s, then the corresponding maximum pressurization pressure of the water jet thruster was 1.89 MPa.

As shown in Figure 3c,d, the experimental obstacle-crossing parameters were in accordance with those of the optimization process. Thus, for the experiment of jumping onto the platform, l_c = 0.615 m and b = 1.04 m; and for the experiment of jumping over the platform, l_c = 0.57 m, a = 0.105 m and b = 0.67 m. The experimental procedure was the same with that of the trajectory accuracy verification experiment. In addition, the robot's experimental jumping trajectories and pitch angle curves were obtained by analysis of experimental video in ProAnalyst and the onboard attitude sensor respectively.

3. Results and Discussion

3.1. Modified Whale Optimization Algorithm

3.1.1. Performance Improvement Effect of Each Improvement Strategy

In order to analyze the performance improvement effect of each improvement strategy, the 10 benchmark functions were used to test each improvement strategy. Therefore, the WOA modified by tent mapping strategy was named MWOA-I; the WOA modified by non-linear convergence factor and self-adaptive weight strategy was named MWOA-II; the WOA modified by diversity variation strategy was named MWOA-III. The comparison results were shown in Table 3.

Table 3. Comparison of performance improvement effect of each improvement strategy.

Benchmark Function	Evaluation Indicator	MWOA-I	MWOA-II	MWOA-III	MWOA
f_1	Mean value	1.05×10^{-73}	**0**	1.54×10^{-174}	**0**
	Standard deviation	4.82×10^{-73}	**0**	0	**0**
f_2	Mean value	1.59×10^{-51}	$\mathbf{6.03 \times 10^{-210}}$	4.19×10^{-134}	1.27×10^{-139}
	Standard deviation	6.30×10^{-51}	**0**	2.92×10^{-133}	6.94×10^{-139}
f_3	Mean value	5.64×10^{4}	**0**	8.36×10^{-176}	**0**
	Standard deviation	1.32×10^{4}	**0**	0	**0**
f_4	Mean value	2.80×10^{1}	2.83×10^{1}	2.86×10^{1}	$\mathbf{2.77 \times 10^{1}}$
	Standard deviation	4.17×10^{-1}	3.20×10^{-1}	2.60×10^{-1}	$\mathbf{2.06 \times 10^{-1}}$
f_5	Mean value	-1.05×10^{4}	-1.22×10^{4}	$\mathbf{-1.25 \times 10^{4}}$	-1.24×10^{4}
	Standard deviation	1.74×10^{3}	1.04×10^{3}	$\mathbf{1.18 \times 10^{2}}$	3.03×10^{2}
f_6	Mean value	1.89×10^{-15}	**0**	**0**	**0**
	Standard deviation	1.04×10^{-14}	**0**	**0**	**0**
f_7	Mean value	3.97×10^{-15}	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$
	Standard deviation	2.59×10^{-15}	$\mathbf{1.00 \times 10^{-31}}$	$\mathbf{1.00 \times 10^{-31}}$	$\mathbf{1.00 \times 10^{-31}}$
f_8	Mean value	3.66×10^{-2}	**0**	**0**	**0**
	Standard deviation	9.24×10^{-2}	**0**	**0**	**0**
f_9	Mean value	2.22	3.29	1.75	**1.76**
	Standard deviation	2.02	3.57	1.87	$\mathbf{8.52 \times 10^{-1}}$
f_{10}	Mean value	7.12×10^{-4}	6.52×10^{-4}	3.98×10^{-4}	$\mathbf{3.72 \times 10^{-4}}$
	Standard deviation	4.75×10^{-4}	1.83×10^{-4}	1.09×10^{-4}	$\mathbf{1.04 \times 10^{-4}}$

The bold column for each benchmark function in Table 3 is the best result. According to Table 3, the MWOA had best performances in 8 out of the 10 benchmark functions, then the MWOA-II and MWOA-III had 6 best results and 4 best results respectively. Therefore, it can be concluded that the non-linear convergence factor and self-adaptive weight strategy had a larger performance improvement effect than the other two strategies. However, the MWOA, which combined the three strategies still outnumber any other algorithms which utilized only one strategy in the best optimization results.

3.1.2. Performance Comparison with Other Intelligent Optimization Algorithms

In order to compare the MWOA's performance with other intelligent optimization algorithms, the 10 benchmark functions were used again to test the performance of WOA, DA, HHO and MWOA. The comparison results were shown in Table 4, and the convergence curves of these four algorithms for f_1 and f_2 were shown Figure 6 (the other convergence curves were presented in Figure A1).

Table 4. Performance comparison of MWOA with other intelligent optimization algorithms.

Benchmark Function	Evaluation Indicator	WOA	DA	HHO	MWOA
f_1	Mean value	3.98×10^{-73}	2.03×10^{3}	4.57×10^{-93}	**0**
	Standard deviation	1.97×10^{-72}	1.10×10^{3}	2.50×10^{-92}	**0**
f_2	Mean value	1.65×10^{-50}	1.44×10^{1}	2.07×10^{-50}	**1.27×10^{-139}**
	Standard deviation	7.67×10^{-50}	5.58	1.09×10^{-49}	**6.94×10^{-139}**
f_3	Mean value	4.16×10^{4}	1.42×10^{4}	1.88×10^{-70}	**0**
	Standard deviation	1.09×10^{4}	8.49×10^{3}	8.12×10^{-70}	**0**
f_4	Mean value	2.79×10^{1}	4.25×10^{5}	**1.32×10^{-2}**	2.77×10^{1}
	Standard deviation	3.35×10^{-1}	3.93×10^{5}	**1.92×10^{-2}**	2.06×10^{-1}
f_5	Mean value	-9.83×10^{3}	-5.36×10^{3}	**-1.26×10^{4}**	-1.24×10^{4}
	Standard deviation	1.92×10^{3}	6.57×10^{2}	**8.14×10^{1}**	3.03×10^{2}
f_6	Mean value	3.31	1.88×10^{2}	**0**	**0**
	Standard deviation	1.82×10^{1}	4.36×10^{1}	**0**	**0**
f_7	Mean value	4.45×10^{-15}	1.05×10^{1}	**8.88×10^{-16}**	**8.88×10^{-16}**
	Standard deviation	2.09×10^{-15}	1.47	**1.00×10^{-31}**	**1.00×10^{-31}**
f_8	Mean value	3.70×10^{-18}	1.86×10^{1}	**0**	**0**
	Standard deviation	2.03×10^{-17}	1.15×10^{1}	**0**	**0**
f_9	Mean value	2.50	**1.23**	1.29	1.76
	Standard deviation	2.91	**4.28×10^{-1}**	9.75×10^{-1}	8.52×10^{-1}
f_{10}	Mean value	5.96×10^{-4}	2.24×10^{-3}	3.78×10^{-4}	**3.72×10^{-4}**
	Standard deviation	2.98×10^{-4}	3.42×10^{-3}	2.27×10^{-4}	**1.04×10^{-4}**

Similarly, the bold column for each benchmark function in Table 4 is the best result. According to Table 4, the MWOA had the 7 best results among the 10 benchmark functions, then the HHO and DA had the 5 best results and 1 of the best results, respectively. Thus, it can be concluded that the convergence precision and stability of the MWOA were the best according to the evaluation indicators of the mean value and standard deviation.

By analyzing the convergence curves of four intelligent optimization algorithms, it can be concluded that the MWOA had the fastest convergence speeds in 6 benchmark functions (f_1,f_2,f_3,f_6,f_7 and f_8), which surpassed the other algorithms. Compared with standard WOA, the MWOA had better convergence speeds in all 10 benchmark functions. For the optimization of f_6, f_7, f_8 and f_{10}, the convergence curves of MWOA showed multiple inflection points while descending, which means its ability of escaping from local optimal solution was obviously improved. Thus, by performance comparison with other intelligent optimization algorithms, the reasonability and effectiveness of MWOA can be proved.

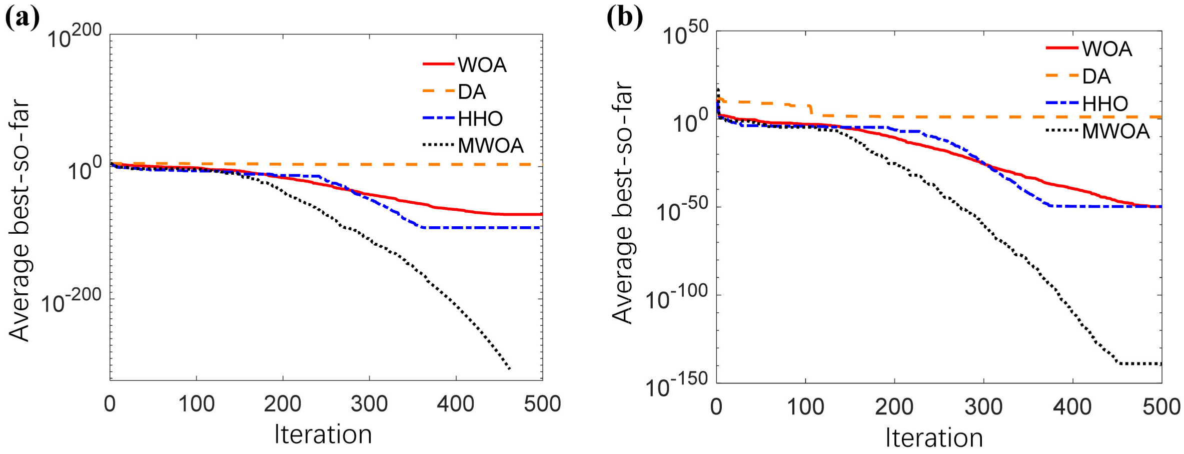

Figure 6. Convergence curves of four intelligent optimization algorithms. (**a**) Convergence curves for f_1. (**b**) Convergence curves for f_2. Average best-so-far means the average of the solution obtained so far in each iteration over 30 runs.

3.1.3. Optimization of Jumping Trajectory

After determining the total fitness function, the centroid jumping trajectory of jumping onto a platform and jumping over a platform can be optimized under different weight coefficient combinations. As shown in Figure 3c,d, the relevant parameter values of jumping onto the platform are: l_c = 0.615 m, l_s = 0.32 m, l_r = 0.30 m and b = 1.04 m; then the relevant parameter values of jumping over platform are: l_c = 0.57 m, l_s = 0.32 m, l_r = 0.30 m, a = 0.105 m and b = 0.67 m. It is noticeable that these obstacle-crossing parameters are predetermined by the experimenters.

Therefore, the searching space was firstly determined by the decision criteria (3) and (5). Then the weight coefficient combinations were determined and the fitness function (29) was optimized by the MWOA. Lastly, the optimization results were listed in Table 5, where optimization I and II were for jumping onto a platform and jumping over a platform, respectively. The corresponding centroid jumping trajectories were shown in Figure 7.

Table 5. Optimization results of jumping trajectory under different weight coefficient combinations.

No.	Priority	Weight Coefficient Combination	Optimization Result I	Optimization Result II
1	Horizontal jumping distance	ω_1 = 0.6132, ω_2 = 0.2125, ω_3 = 0.1743	φ_0 = 71.2°, v_0 = 8 m/s	φ_0 = 62.5°, v_0 = 8 m/s
2	Vertical jumping distance	ω_1 = 0.0315, ω_2 = 0.8149, ω_3 = 0.1536	φ_0 = 81.9°, v_0 = 8 m/s	φ_0 = 81.9°, v_0 = 8 m/s
3	Landing safety	ω_1 = 0.1852, ω_2 = 0.0926, ω_3 = 0.7222	φ_0 = 76.0°, v_0 = 5.57 m/s	φ_0 = 76.0°, v_0 = 5.57 m/s

Figure 8a,b show the optimum take-off parameters under the third weight coefficient combinations in Table 5 for jumping onto the platform and jumping over the platform, respectively. It can be seen that the optimum take-off parameters for jumping onto the platform and jumping over the platform are both φ_0 = 76.0°, v_0 = 5.57 m/s in their separate searching spaces, thus the reasonability of optimization results of theoretical jumping trajectory can be verified.

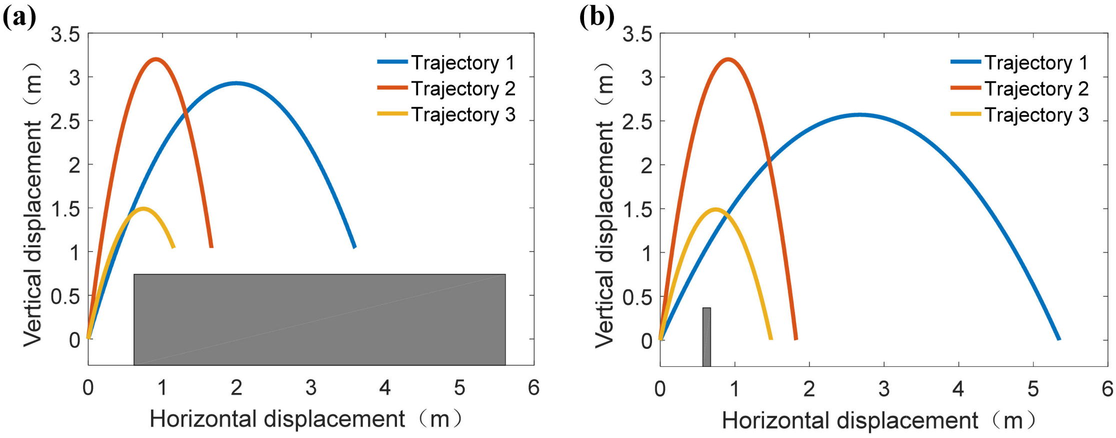

Figure 7. The optimal jumping trajectory under different weight coefficient combinations. (**a**) The optimal centroid jumping trajectories of jumping robot while jumping onto platform. (**b**) The optimal jumping centroid trajectories of jumping robot while jumping over the platform.

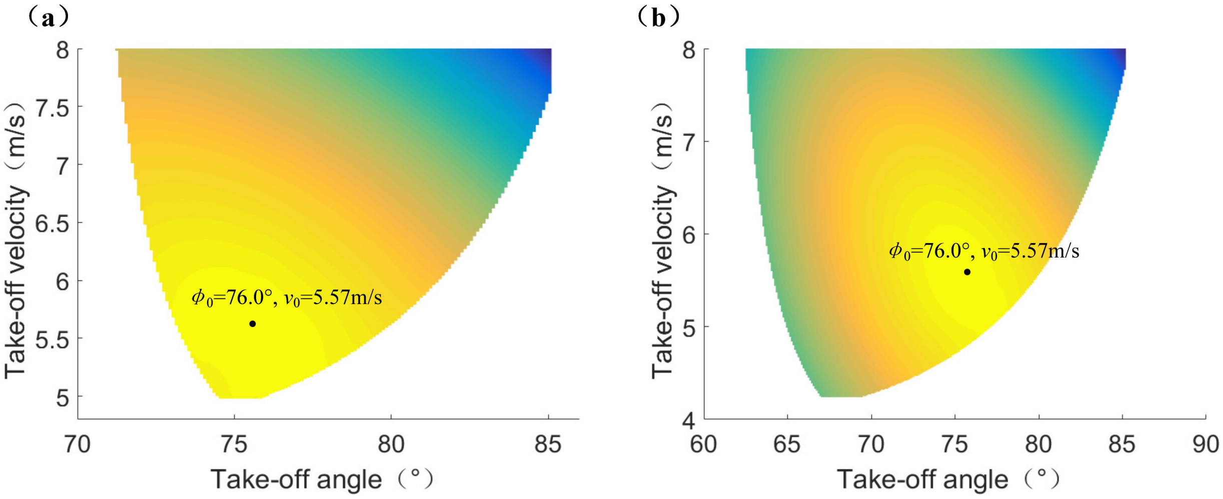

Figure 8. The optimization results of theoretical jumping trajectory. (**a**) The optimum take-off parameters for jumping onto the platform. (**b**) The optimum take-off parameters for jumping over the platform.

3.2. Aerial Pitch Control Experiments

Figure 9 is a representative time sequence of aerial pitch control experiments (Experiment No. 4, Group 1). Figure 10 shows the jumping robot's pitch angle curves of the aerial pitch control experiment, each curve started at the moment when the robot began to fall and ended at the moment when it contacted the ground. It can be seen from Figure 9 that the attitude sensor module detected the robot's acceleration variation when the robot started falling freely, thus the assistant legs were driven to rotate a theoretical angle $\Delta\theta_t = -48.5°$. During the rotation of the assistant legs, the main body's pitch angle also increased from the initial value −17.3°. According to Figure 10a, the robot's pitch angle reached 0° at 422 ms and then 1.81° at 531 ms when the robot contacted the ground. After landing, the assistant legs reset to their initial position (parallel to the central axis of the water jet thruster), and the aerial pitch control process was completed.

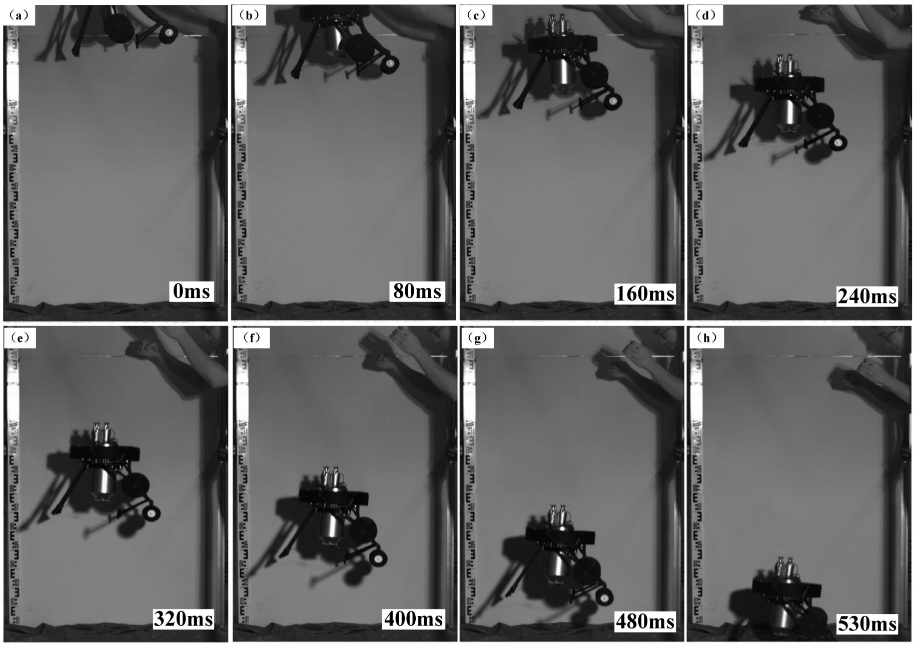

Figure 9. The time sequence of aerial pitch control experiment No. 4, Group 1.

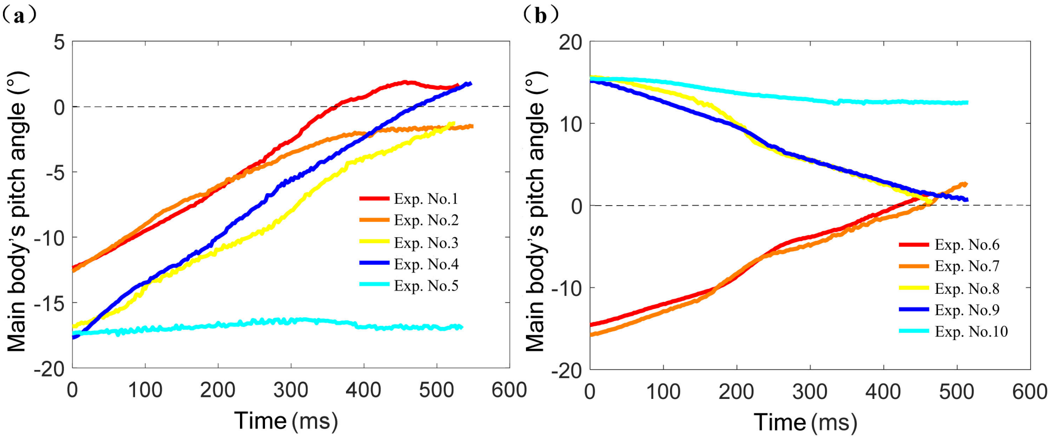

Figure 10. The jumping robot's pitch angle curves of aerial pitch control experiment. (**a**) The pitch angle curves of Group 1. (**b**) The pitch angle curves of Group 2.

As the expected value of pitch angle is 0°, it can be known from Figure 10 that the absolute values of maximum error of pitch angle in Group 1 and Group 2 were 1.81° (Experiment No. 4) and 2.79° (Experiment No. 6) respectively. However, the pitch angle did not change obviously in Experiment No. 5 and Experiment No. 10 which did not conduct aerial pitch control movements.

The RMSE (Root Mean Square Error) [34] of the pitch angle curves of Experiment No. 4, Experiment No. 5, Experiment No. 9, and Experiment No. 10 can be compared respectively to verify the effectiveness of aerial pitch control movement. Hence the calculation results of RMSE for Experiment No. 4, No. 5, No. 9, and No. 10 were: $RMSE_4 = 9.227°$, $RMSE_5 = 16.821°$, $RMSE_9 = 8.637°$, $RMSE_{10} = 13.379°$, respectively. If it is obvious that $RMSE_4 < RMSE_5$ and $RMSE_9 < RMSE_{10}$, then the effectiveness of the mathematical model of aerial pitch control movement can be verified and it can be applied in the actual jumping and obstacle-crossing experiment for the jumping robot.

3.3. Jumping Experiments of the Robot

3.3.1. Trajectory Accuracy Verification Experiments

After the outdoor terrestrial jumping experiment, the four experimental jumping trajectories can be obtained and their comparisons with corresponding theoretical jumping trajectories were shown in Figure 11. Therefore, the RMSE of each experimental jumping trajectory can be calculated to evaluate its accuracy. The calculation results of RMSE for Figure 11a–d were: $\text{RMSE}_a = 0.057$ m, $\text{RMSE}_b = 0.095$ m, $\text{RMSE}_c = 0.074$, $\text{RMSE}_d = 0.071$ m, respectively. The maximum average degree of deviation in vertical displacement was 0.095 m, thus it can be concluded that the error between experimental jumping trajectory and theoretical jumping trajectory was small.

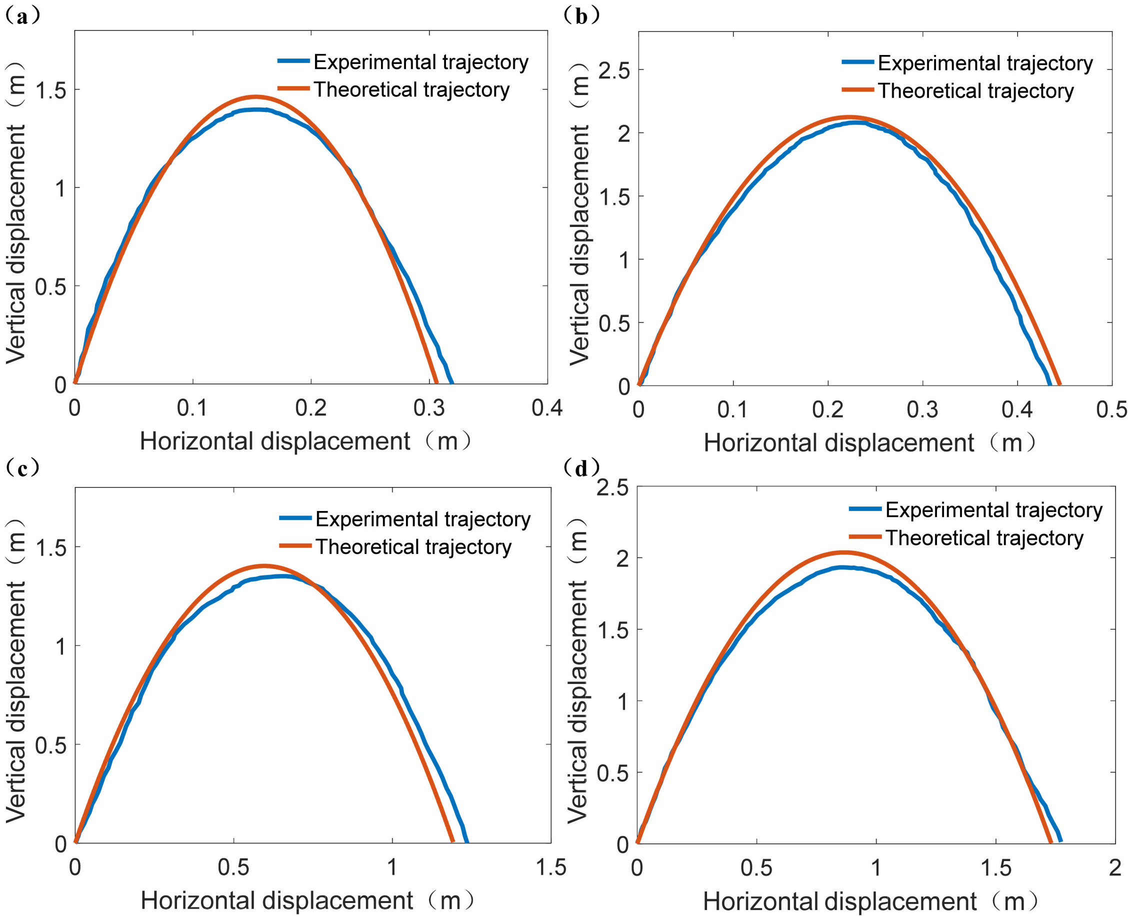

Figure 11. The comparisons of experimental jumping trajectories and theoretical jumping trajectories under different take-off parameters. (**a**) $\varphi_0 = 87°$, $P_{\max} = 1.81$ MPa. (**b**) $\varphi_0 = 87°$, $P_{\max} = 2.23$ MPa. (**c**) $\varphi_0 = 78°$, $P_{\max} = 1.81$ MPa. (**d**) $\varphi_0 = 78°$, $P_{\max} = 2.23$ MPa.

3.3.2. Obstacle-Crossing Experiments

The time sequence of jumping robot's process of jumping onto platform and jumping over platform are shown in Figures 12 and 13 respectively (The whole obstacle-crossing jumping processes are presented in https://www.youtube.com/watch?v=op6UnGVcYnw, (accessed on 30 March 2021) and in Video S1 (Supplementary Materials)). It can be known that the robot did successfully jump over and jump onto the platform with given take-off parameters. Although the robot did not compensate well in its roll direction due to the lack of attitude control action in the roll direction, the assistant legs' rotation did have an obvious adjusting effect on the robot main body's pitch angle. Then the jumping robot's jumping trajectory comparisons and pitch angle curves of the main body are shown in Figure 14.

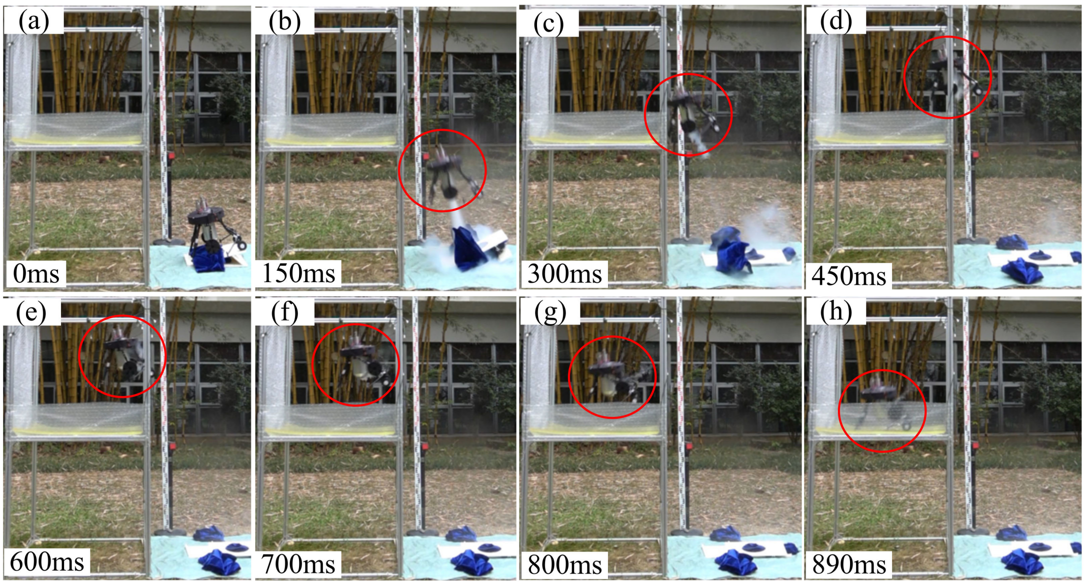

Figure 12. The time sequence of the jumping robot's process of jumping onto the platform.

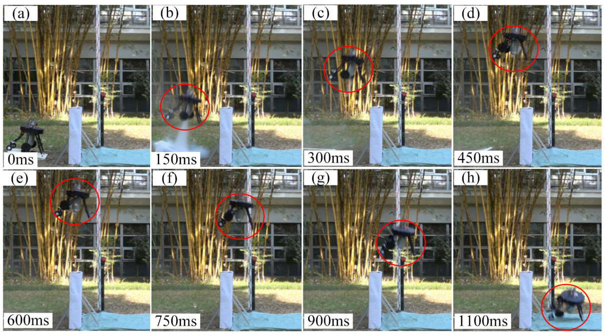

Figure 13. The time sequence of jumping robot's process of jumping over platform.

The comparison of the jumping robot's centroid trajectory of the experiment of jumping onto the platform and jumping over the platform are shown in Figure 14a,b. It can be calculated that the RMSE values of these two experiments are RMSE_a = 0.0537 m and RMSE_b = 0.0731 m, respectively, which are both within the reasonable range. Thus, the accuracies of the robot's jumping trajectory are acceptable.

According to Figure 14c, the robot's pitch angle was stable at around $-14°$ in the initial stage. Then the main body's pitch angle started approaching the expected value of 0° in the effect of the assistant legs' rotation after t = 550 ms. Lastly, the main body's pitch angle increased to $-2.45°$ at t = 890 ms when the robot landed on the platform. Thus, the duration and the absolute value of error of aerial pitch control was 340 ms and 2.45°, respectively. Similarly, in Figure 14d, the robot's pitch angle was also stable at around $-14°$ initially. Then the main body's pitch angle started approaching the expected value of 0° after t = 290 ms. However, the pitch angle did not change obviously after t = 820 ms, and the final pitch angle was $-2.79°$ at t = 1100 ms when the robot landed on the platform. Thus, the duration and the absolute value of error of aerial pitch control was 530 ms and 2.79° respectively.

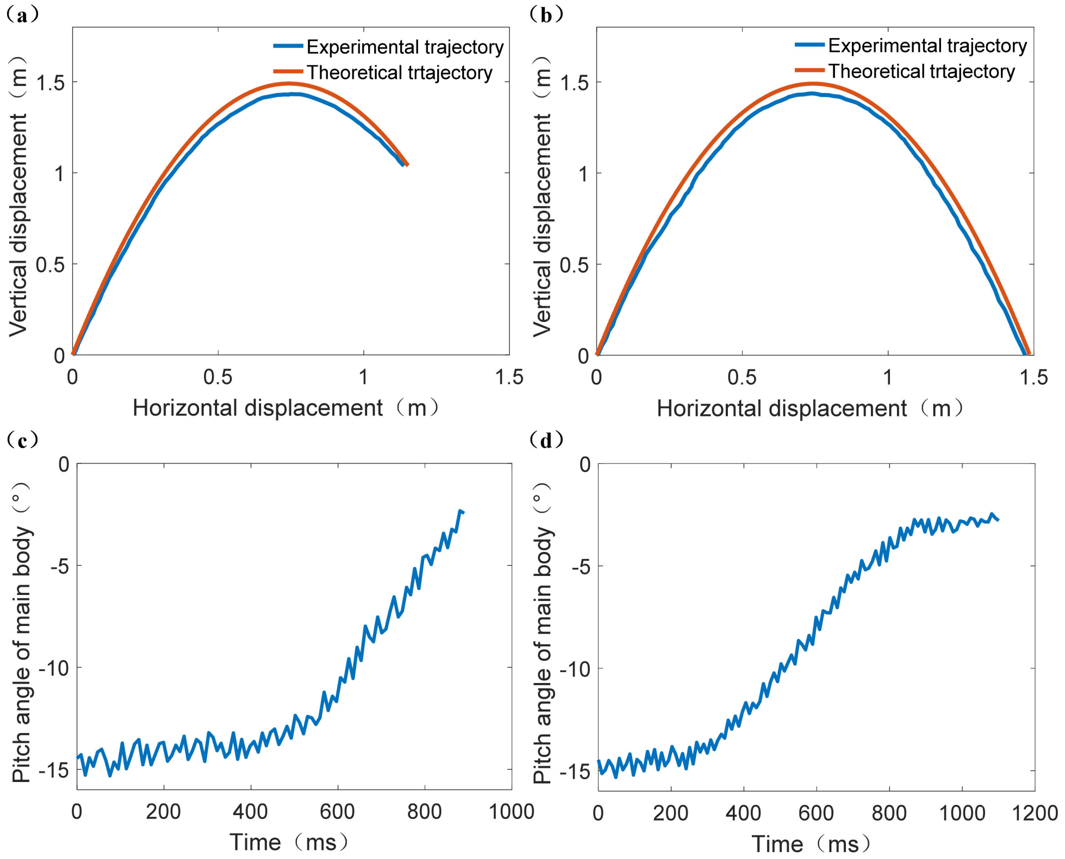

Figure 14. The jumping robot's jumping trajectories and pitch angle curves of the obstacle-crossing experiment. (**a**) The comparison of jumping robot's centroid trajectory of the experiment of jumping onto the platform. (**b**) The comparison of the jumping robot's centroid trajectory of the experiment of jumping over the platform. (**c**) The pitch angle curve of the experiment of jumping onto platform. (**d**) The pitch angle curve of the experiment of jumping over platform. Both the desired pitch angle of main body in (**c**, **d**) are 0°.

Additionally, it can be calculated from Figure 14a that the initial experimental take-off angle and take-off velocity were φ_{0e1} = 75.54° and v_{0e1} = 5.15 m/s respectively for the experiment of jumping onto the platform, and therefore the corresponding fitness value was f_{e1} = 0.8328. Similarly, the calculation results for the experiment of jumping over the platform were φ_{0e2} = 75.53° and v_{0e2} = 5.24 m/s, respectively, therefore the corresponding fitness value was f_{e2} = 0.8337. The absolute value of relative error of fitness value for the experiment of jumping onto the platform and jumping over the platform were e_{f1} = 0.23%, e_{f2} = 0.12%. Therefore, the aerial pitch control model's reasonability and the jumping robot's accurate obstacle-crossing ability can be verified by these two obstacle-crossing experiments.

4. Conclusions

In this paper, a novel jumping robot was devised and the process of how it completed actual obstacle-crossing tasks and aerial pitch control movements are presented. Firstly, the decision criteria for overcoming general obstacles were proposed to obtain the feasible jumping trajectory set for the jumping robot. In order to optimize the jumping trajectory according to the fitness function value, a modified whale optimization algorithm (MWOA) was put forward and its performance superiority was verified by comparing it with other different intelligent optimization algorithms. Then the dynamics model of aerial pitch control was established and the jumping robot's aerial pitch control experiment was conducted. The experimental results demonstrated the effectiveness of the proposed dynamics model. In addition, the robot's trajectory accuracy verification experiment showed that the robot's jumping trajectory was accurate and reliable. Finally, the robot's obstacle-crossing experiments were conducted and the experimental results verified the robot's good ability of obstacle-crossing and aerial pitch control. Collectively, these achievements provide meaningful guidance for the jumping robot's performance in dealing with complex obstacle-crossing tasks and future research on jumping robot's reliability as well as environmental adaptability.

Supplementary Materials: The following are available online at https://www.youtube.com/watch?v=op6UnGVcYnw, Video S1: HITSZ_Jumping robot (accessed on 30 March 2021) and at https://www.mdpi.com/article/10.3390/s21072432/s1, Video S1: HITSZ_Jumping robot.

Author Contributions: Literature search, J.M. and Z.Y.; figures, J.M.; study design, J.M., Z.Y., B.L., F.X. and Y.L.; data collection, J.M., Z.Y., B.L., F.X. and Y.L.; data analysis and interpretation, J.M., Z.Y., and Y.L.; writing—original draft preparation, J.M. and Z.Y.; writing—review and editing, J.M. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2019B090915001, in part by the National Key Research and Development Program of China under Grant 2017YFB1302200, and in part by the Shenzhen Research and Development Program of China under Grant JSGG20180504170011639 and Grant JCYJ20200109112818703. Potsdam under the Geo.X Young academy program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. The benchmark functions of algorithm.

Name	Formula	Searching Range	Dimension	Minimum Value
Sphere	$f_1(x) = \sum_{i=1}^{\dim} x_i^2$	[−100,100]	30	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{\dim} \left\| x_i \right\| + \prod_{i=1}^{\dim} \|x_i\|$	[−10,10]	30	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{\dim} \left(\sum_{j=1}^{i} x_j \right)^2$	[−100,100]	30	0
Rosenbrock	$f_4(x) = \sum_{i=1}^{\dim-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−30,30]	30	0
Schwefel 2.26	$f_5(x) = \sum_{i=1}^{\dim} -x_i \sin(\sqrt{\|x_i\|})$	[−500,500]	30	−12,569.5
Rastrigin	$f_6(x) = \sum_{i=1}^{\dim} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	30	0
Ackley	$f_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{\dim}\sum_{i=1}^{\dim} x_i^2}\right) - \exp(\frac{1}{\dim}\sum_{i=1}^{\dim}\cos(2\pi x_i)) + 20 + e$	[−32,32]	30	0
Griewank	$f_8(x) = \frac{1}{4000}\sum_{i=1}^{\dim} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[−600,600]	30	0
Shekel	$f_9(x) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}]^{-1}$	[−65,65]	2	1
Kowalik	$f_{10}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[−5,5]	4	0.00030

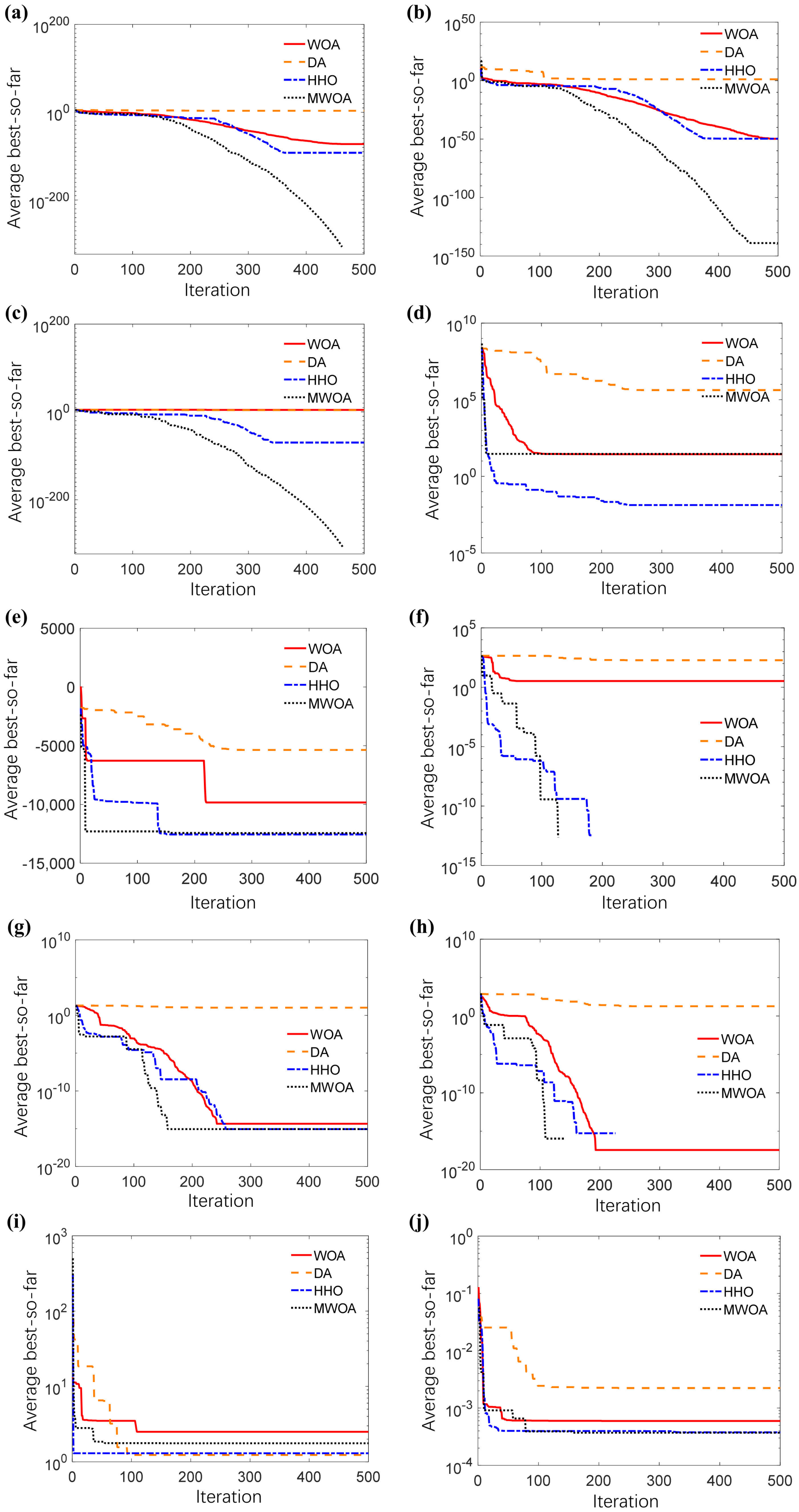

Figure A1. Convergence curves of four intelligent optimization algorithms. (**a**) Convergence curves for f_1. (**b**) Convergence curves for f_2. (**c**) Convergence curves for f_3. (**d**) Convergence curves for f_4. (**e**) Convergence curves for f_5. (**f**) Convergence curves for f_6. (**g**) Convergence curves for f_7. (**h**) Convergence curves for f_8. (**i**) Convergence curves for f_9. (**j**) Convergence curves for f_{10}.

References

1. Alexander, R.M.N. *Principles of Animal Locomotion*; Princeton University Press: Princeton, NJ, USA, 2013; ISBN 9781400849512.
2. Stoeter, S.; Papanikolopoulos, N. Autonomous Stair-Climbing with Miniature Jumping Robots. *IEEE Trans. Syst. Man. Cybern. B Cybern.* **2005**, *35*, 313–325. [CrossRef] [PubMed]
3. Fischer, G.J.; Spletzer, B.L. Long-range hopping mobility platform. *Unmanned Gr. Veh. Technol. V* **2003**, *5083*, 83. [CrossRef]
4. Fischer, G.J. Wheeled hopping mobility. In Proceedings of the European Symposium on Optics & Photonics for Defence & Security, Bruges, Belgium, 26 October 2005; p. 59860H.
5. Ackerman, E. Boston Dynamics Sand Flea Robot Demonstrates Astonishing Jumping Skills. Available online: https://spectrum.ieee.org/automaton/robotics/military-robots/boston-dynamics-sand-flea-demonstrates-astonishing-jumping-skills (accessed on 28 January 2021).
6. Liu, G.H.; Lin, H.Y.; Lin, H.Y.; Chen, S.T.; Lin, P.C. A bio-inspired hopping kangaroo robot with an active tail. *J. Bionic Eng.* **2014**, *11*, 541–555. [CrossRef]
7. Wang, H.; Luan, Y.; Oetomo, D.; Wang, Z. Design, Analysis and Experimental Evaluation of a Gas-Fuel-Powered Actuator for Robotic Hoppers. *IEEE/ASME Trans. Mechatron.* **2015**, *20*, 264–2275. [CrossRef]
8. Zaitsev, V.; Gvirsman, O.; Ben Hanan, U.; Weiss, A.; Ayali, A.; Kósa, G. A locust-inspired miniature jumping robot. *Bioinspir. Biomim.* **2015**, *10*, 66012. [CrossRef] [PubMed]
9. Weiss, A.; Zaitsev, V.; Nabi, N.; Hanan, U. Ben. Landing recovery and orientation control of a locust-inspired miniature jumping robot. *Eng. Res. Express* **2020**, *2*. [CrossRef]
10. Yu, J.; Su, Z.; Wu, Z.; Tan, M. Development of a Fast-Swimming Dolphin Robot Capable of Leaping. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2307–2316. [CrossRef]
11. Yan, J.; Wang, T.; Zhang, X.; Zhao, J. Structural design and dynamic analysis of biologically inspired water-jumping robot. In Proceedings of the 2014 IEEE International Conference on Information and Automation, Hailar, China, 28–30 July 2014; pp. 1295–1299.
12. Siddall, R.; Kovač, M. A water jet thruster for an aquatic micro air vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 3979–3985.
13. Zufferey, R.; Ancel, A.O.; Farinha, A.; Siddall, R.; Armanini, S.F.; Nasr, M.; Brahmal, R.V.; Kennedy, G.; Kovac, M. Consecutive aquatic jump-gliding with water-reactive fuel. *Sci. Robot.* **2019**, *4*. [CrossRef] [PubMed]
14. Siddall, R.; Kovač, M. Launching the AquaMAV: Bioinspired design for aerial-aquatic robotic platforms. *Bioinspir. Biomim.* **2014**, *9*, 31001. [CrossRef] [PubMed]
15. Jusufi, A.; Goldman, D.I.; Revzen, S.; Full, R.J. Active tails enhance arboreal acrobatics in geckos. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 4215–4219. [CrossRef] [PubMed]
16. Libby, T.; Moore, T.Y.; Chang-Siu, E.; Li, D.; Cohen, D.J.; Jusufi, A.; Full, R.J. Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature* **2012**, *481*, 181–186. [CrossRef] [PubMed]
17. Jusufi, A.; Kawano, D.T.; Libby, T.; Full, R.J. Righting and turning in mid-air using appendage inertia: Reptile tails, analytical models and bio-inspired robots. *Bioinspir. Biomim.* **2010**, *5*, 45001. [CrossRef] [PubMed]
18. Johnson, A.M.; Libby, T.; Chang-Siu, E.; Tomizuka, M.; Full, R.J.; Koditschek, D.E. Tail assisted dynamic self righting. In Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies, Baltimore, MD, USA, 23–26 July 2012; pp. 611–620.
19. Chang-Siu, E.; Libby, T.; Tomizuka, M.; Full, R. A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1887–1894.
20. Kohut, N.J.; Pullin, A.O.; Haldane, D.W.; Zarrouk, D.; Fearing, R.S. Precise dynamic turning of a 10 cm legged robot on a low friction surface using a tail. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 3299–3306.
21. Mo, J.; Miao, Z.; Li, B.; Zhang, Y.; Song, Z. Design, analysis, and performance verification of a water jet thruster for amphibious jumping robot. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 5431–5447. [CrossRef]
22. Mo, J.; Xiao, J.; Lin, F.; Li, Y.; Song, Z. Study of a new water jet thruster for robotic returnable jumping. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**. [CrossRef]
23. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
24. Shan, L.; Qiang, H.; Li, J.; Wang, Z.Q. Chaotic optimization algorithm based on Tent map. *Kongzhi yu Juece/Control Decis.* **2005**, *20*, 179–182.
25. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
26. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
27. Yang, Y.L.; Chao, P.C.P.; Sung, C.K. Landing posture control for a generalized twin-body system using methods of input-output linearization and computed torque. *IEEE/ASME Trans. Mechatron.* **2009**, *14*, 326–336. [CrossRef]

28. Agrawal, S.K.; Zhang, C. An approach to posture control of free-falling twin bodies using differential flatness. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, Taipei, Taiwan, China, 18–22 October 2010; pp. 685–690.
29. Chang-Siu, E.; Libby, T.; Brown, M.; Full, R.J.; Tomizuka, M. A nonlinear feedback controller for aerial self-righting by a tailed robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 32–39.
30. Li, Z.; Montgomery, R. Dynamics and optimal control of a legged robot in flight phase. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; pp. 1816–1821.
31. Zhao, J.; Xu, J.; Gao, B.; Xi, N.; Cintron, F.J.; Mutka, M.W.; Xiao, L. MSU jumper: A single-motor-actuated miniature steerable jumping robot. *IEEE Trans. Robot.* **2013**, *29*, 602–614. [CrossRef]
32. Gu, E.Y.L. *Robotic Dynamics: Modeling and Formulations*; Springer: Berlin, Germany, 2013; pp. 231–291. ISBN 978-3-642-39046-3.
33. Chapra, S.C. *Applied Numerical Methods with MATLAB for Engineers and Scientists*; McGraw-Hill Science/Engineering/Math: New York, NY, USA, 2006; ISBN 9780874216561.
34. Heumann, C.; Schomaker, M.; Shalabh. *Introduction to Statistics and Data Analysis*; Springer International Publishing: Berlin, Germany, 2016; ISBN 9783319461625.