



Article

Controlling a One-Legged Robot to Clear Obstacles by Combining the SLIP Model with Air Trajectory Planning

Senwei Huang  and Xiuli Zhang *

School of Mechanical, Electronic and Control Engineering, Beijing Jiaotong University, Beijing 100044, China

* Correspondence: zhangxl@bjtu.edu.cn

Abstract: Legged animals can adapt to complex terrains because they can step or jump over obstacles. Their application of foot force is determined according to the estimation of the height of an obstacle; then, the trajectory of the legs is controlled to clear the obstacle. In this paper, we designed a three-DoF one-legged robot. A spring-loaded inverted pendulum model was employed to control the jumping. Herein, the jumping height was mapped to the foot force by mimicking the jumping control mechanisms of animals. The foot trajectory in the air was planned using the Bézier curve. Finally, the experiments of the one-legged robot jumping over multiple obstacles of different heights were implemented in the PyBullet simulation environment. The simulation results demonstrate the effectiveness of the method proposed in this paper.

Keywords: legged robot; SLIP; foot trajectory planning; Bézier curve; bionic hopping control



Citation: Huang, S.; Zhang, X. Controlling a One-Legged Robot to Clear Obstacles by Combining the SLIP Model with Air Trajectory Planning. *Biomimetics* **2023**, *8*, 66. <https://doi.org/10.3390/biomimetics8010066>

Academic Editor: Xuechao Chen

Received: 29 November 2022

Revised: 3 February 2023

Accepted: 3 February 2023

Published: 5 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Legged robots are a research hotspot in the field of robotics and have achieved excellent performance in structural design and motion control [1–10]. The one-legged robot is a special kind of legged robot because it is the basis of legged robot research. Most technologies of multi-legged robots are extended from one-legged robots. Therefore, the breakthrough of one-legged-robot control technology is the key to promoting the development of legged robots. Research on the hopping control of single-legged robots has been conducted for more than 40 years and has also yielded excellent research results. Raibert decomposed the hopping process into three parts and realized the continuous hopping of a physical one-legged robot [11]. Inspired by the structure of human joints, a one-legged robot with a controllable reduction ratio was created, achieving a 42 cm vertical jump [12]. Salto-1P, an untethered monopodal robot, can hop on discontinuous surfaces such as office furniture and track a moving platform. Salto-1P's deadbeat foot placement jumping controller allows it to precisely position the foot even on trajectories with abrupt changes in height, speed, and direction [13]. A parallel one-legged robot with a brushless direct current motor and a harmonic driver uses a virtual model control (VMC) approach to jump and land with high dynamic motion [14]. Moreover, the efficiency of virtual model control has been validated for one-legged robots [15,16]. There are other ways to achieve legged robot hopping, such as ZMP [17,18], MPC [19,20], WBC [21–23], and trajectory optimization-based methods [24–26]. Moreover, artificial intelligence algorithms, such as reinforcement learning and deep reinforcement learning, are also applied to jumping control in legged robots [27,28]. In these works, the researchers focused on the structural design, jumping height, running speed, and stability control of the one-legged robot. The performance of various control methods is also closely related to the robot structure and control objectives. In addition, some approaches need to be extended to the full locomotion task. The advantage of a one-legged robot is its exceptional ability to jump and clear obstacles. However, few studies have covered the obstacle clearing of the one-legged robot. On the other hand, trajectory planning is also extremely important for one-legged robots and has been ignored by researchers.

Jumping is an important locomotion strategy employed by animals such as frogs, kangaroos, and galagos. Compared with walking and running, jumping can better adapt animals to unstructured environments and even allow them to overcome obstacles several times larger than their body size [29,30]. By using mathematical models to study the muscle properties, leg conformation, and jumping technique of humans, frogs, locusts, and other animals during jumping, it was found that jump height increases with the increase in isometric force exerted by the leg muscles, maximum shortening velocity, and a series of consistencies [31]. Animals can predict their jumping range according to their weight, leg length, and the amount of energy they can generate, so as to assess whether they can cross obstacles in the environment [32]. Legged animals such as dogs, kangaroos, and locusts plan their foot trajectory and jump height according to the shape of the obstacle [33,34]. The body attitude of the frog hopper in jumping is set by the front and middle legs, and the attitude is adjusted by the extension of the hind legs to control the direction of movement after take-off [35]. Humans can cross obstacles by controlling the trajectory of their bodies as well as the amount of foot force. Furthermore, animals can perceive ground information through the visual and tactile systems and then plan movement trajectories according to the size and location of obstacles [36,37]. The animal control of body trajectory and foot force is what the robot should learn in order to improve motion performance. Engineering practice has also found that the robot finds it difficult to walk over obstacles with compliant behaviors, because certain motions such as the lifting of feet are needed. However, environment perception and trajectory planning are important problems that jumping robots still have to solve before they can leave the laboratory and be applied in real unstructured environments. Therefore, one research focus in bionic legged robots is to explore the jumping mechanisms of animals and imitate their behavior. Biologists have summarized the dynamic motion of animals using the spring-loaded inverted pendulum (SLIP) model, which is commonly used in hopping control in legged robots [38]. In the SLIP model, the mass of the robot is centralized in the center of the body, and the legs act as a massless spring. The jumping mechanism of locusts well reflects the SLIP model, where the movement of the locust legs during jumping acts like a spring to store and release energy [39]. Moreover, biological research has revealed that the stiffness of animal legs changes dynamically [40,41]. This also explains why it is difficult to obtain good performance when using fixed SLIP model spring stiffness to describe the dynamic characteristics of robot legs. To address this issue, methods for qualitatively analyzing the change in spring stiffness with the increase in running speed have been proposed [42,43].

In this paper, we present an obstacle-clearing method for a bio-inspired one-legged hopping robot that combines the SLIP model with air trajectory planning. The inspiration comes from planning the body trajectory and the magnitude of the foot force according to the height and shape of the obstacles when a human or an animal performs a high jump. To verify the effectiveness of the proposed control method and estimate the jumping performance of the one-legged robot, a simulation experiment was designed to control a one-legged robot continuously jumping over several obstacles of different heights. The experimental results show that the controller designed in this paper can realize one-legged-robot jumping and obstacle-clearing control, which verifies that the control strategy proposed in this paper is effective. The novelty and main contribution of this paper are to propose a method for incorporating Bézier curve-based foot trajectory planning into SLIP model control to imitate the jumping and obstacle-clearing mechanism of animals. It also proposes a method to supplement the energy loss of the robot according to the relationship between the foot force and the speed of the animal when running.

The remainder of the paper is organized as follows: In Section 2, the one-legged robot structure and kinematics are presented. Section 3 introduces foot trajectory planning based on the Bézier curve. Section 4 introduces the control method and reports the building procedure of the controller of the one-legged robot. Section 5 presents the obstacle-clearing experiment and analyzes the results. Finally, Section 6 concludes the paper and discusses future work.

2. One-Legged Robot

2.1. Structural Design

By mimicking the legs of quadrupeds, we built a one-legged robot using PyBullet simulation software. The robot was composed of a body, a femur, a tibia, and a foot, as shown in Figure 1a. The robot had three degrees of freedom: hip roll, hip pitch, and knee pitch. The foot was fixed at the lower end of the tibia. The model parameters are listed in Table 1. The robot's dimensions were $0.3 \text{ m} \times 0.3 \text{ m} \times 0.8 \text{ m}$, and it weighed 130 kg.

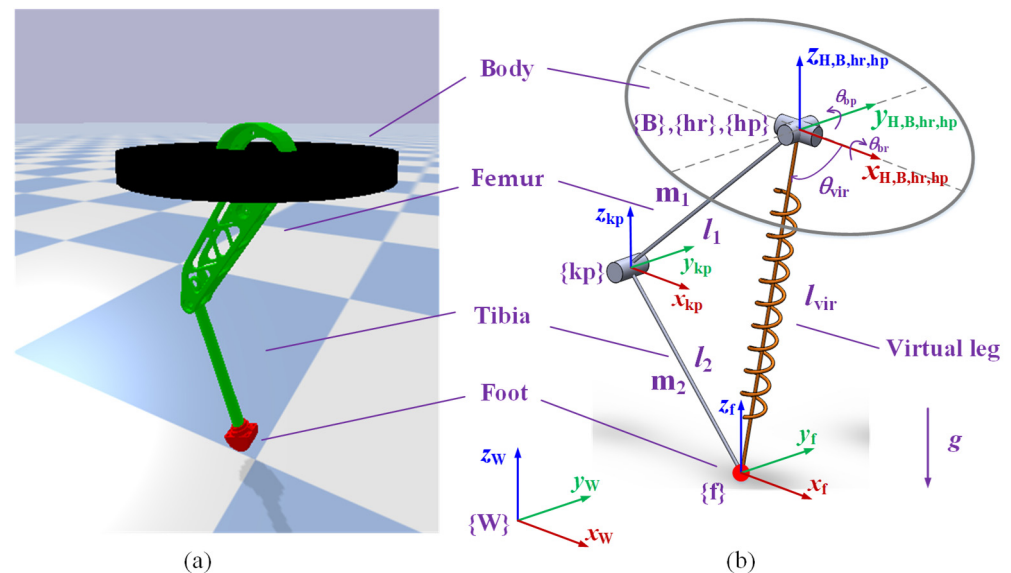


Figure 1. Model of the one-legged robot. (a) Three-dimensional model. (b) Kinematic model.

Table 1. Model parameters of the one-legged robot.

Parameter	Value	Definition
m_1	6 kg	Femur link mass
m_2	4 kg	Tibia link mass
l_1	0.4 m	Femur link length
l_2	0.35 m	Tibia link length
θ_1	/	Hip joint roll angle
θ_2	/	Hip joint pitch angle
θ_3	/	Knee joint pitch angle
l_{vir}	/	Virtual leg length
θ_{vir}	/	Virtual leg pitch angle
θ_{br}	/	Body roll angle
θ_{bp}	/	Body pitch angle

2.2. Kinematic Model

The coordinate frames of the one-legged robot are shown in Figure 1b. {W} is the world coordinate frame with the origin fixed at the center of the ground. {B} is the body coordinate frame with the origin fixed at the geometric center of the body. {H} is the reference coordinate frame for foot trajectory planning, with the same origin as {B} and the axes always oriented in the same direction as {W}. {hr} and {hp} are the hip joint roll and pitch coordinate frame, respectively, with the origin fixed at the geometric center of the hip joint. {kp} is the knee joint pitch coordinate frame with the origin fixed at the geometric center of the knee joint. {f} is the foot coordinate frame with the origin fixed at the geometric center of the foot. The x-axis of all coordinate frames points in the direction of robot movement; the z-axis of all coordinate frames is vertically upward-oriented; and the y-axis of all coordinate frames is determined according to the right-hand rule. The kinematics model of the one-legged robot was established based on the D-H method. According to the

forward kinematics, the position of the foot in the {B} coordinate frame can be represented as

$$\begin{cases} {}^B x_f = l_1 c_2 + l_2 c_{23} \\ {}^B y_f = (l_1 s_2 + l_2 s_{23}) s_1 \\ {}^B z_f = -(l_1 s_2 + l_2 s_{23}) c_1 \end{cases} \quad (1)$$

where $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, $s_{ij} = \sin(\theta_i + \theta_j)$, $c_{ij} = \cos(\theta_i + \theta_j)$, and $i, j = 1, 2, 3 \ i \neq j$. The inverse kinematics are

$$\begin{cases} \theta_1 = -\arctan \frac{{}^B y_f}{{}^B z_f} \\ \theta_2 = \arctan \left(\frac{l_1 + c_3 l_2}{s_3 l_2} \right) - \arctan \left(\frac{{}^B x_f}{\pm \sqrt{l_{vir}^2 - {}^B x_f^2}} \right) \\ \theta_3 = \arccos \frac{l_{vir}^2 - l_1^2 - l_2^2}{2 l_2 l_3} \end{cases} \quad (2)$$

where $l_{vir} = \sqrt{{}^B x_f^2 + {}^B y_f^2 + {}^B z_f^2}$ is the virtual leg length.

The partial differential of Equation (1) is calculated to obtain the foot Jacobian matrix from the foot to the body coordinate frame as follows:

$$\mathbf{J} = \begin{bmatrix} 0 & -l_1 s_2 - l_2 s_{23} & -l_2 s_{23} \\ c_1(l_1 s_2 + l_2 s_{23}) & s_1(l_1 c_2 + l_2 c_{23}) & l_2 s_1 c_{23} \\ s_1(l_1 s_2 + l_2 s_{23}) & -c_1(l_1 c_2 + l_2 c_{23}) & -l_2 c_1 c_{23} \end{bmatrix} \quad (3)$$

3. Foot Trajectory Planning Based on the Bézier Curve

There are numerous methods for planning foot trajectory, such as the rectangular planning method [44], the elliptical planning method [45], the composite cycloid-based approach [46], the composite trajectory planning method [47], the polynomial-based method [48], and the Bézier curve-based approach [49]. Compared with other foot trajectory planning methods, the Bézier curve-based method has the following advantages: (1) the Bézier curve changes smoothly from the starting point to the end point due to its continuous higher-order derivatives, which can ensure continuous and smooth motion; (2) the derivative of the n -order Bézier curve is the $n-1$ -order Bézier curve, so the velocity and acceleration of the foot can be planned; (3) the shape of the curve is determined by a relatively small number of control points; (4) because the first and last control points of the curve are on the curve, it is possible to splice multiple Bézier curves to achieve a bionic foot trajectory with more diverse shapes. Therefore, this paper uses the Bézier curve to generate and parameterize the foot trajectory of the swing phase. A Bézier curve of degree n can be represented as

$$C(u) = \sum_{i=0}^n B_{n,i}(u) P_i \quad (4)$$

where $B_{n,i}(u)$ is a Bernstein polynomial as defined by $B_{n,i}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$, $u \in [0, 1]$ is the Bézier curve parameter, and P_i are the control points.

The time parameter is introduced to organize the time of the foot trajectory.

$$u = at, \quad t \in [0, t_{\max}] \quad (5)$$

where t_{\max} is the time it takes the foot to move from the start to the end of the trajectory and a is the time coefficient given by $a = 1/t_{\max}$.

The speed of foot movement can be obtained by calculating the first derivative of the foot trajectory.

$$\dot{C}(at) = \sum_{i=0}^{n-1} B_{n-1,i}(at) [na(P_{i+1} - P_i)] \quad (6)$$

The acceleration of foot movement can be obtained by calculating the second derivative of the foot trajectory.

$$\ddot{C}(at) = \sum_{i=0}^{n-2} B_{n-2,i}(at) \left[n(n-1)a^2(P_{i+2} - 2P_{i+1} + P_i) \right] \quad (7)$$

According to Equations (4), (6), and (7), it is simple to acquire

$$\begin{cases} C(0) = P_0 \\ \dot{C}(0) = na(P_1 - P_0) \\ \ddot{C}(0) = na(P_2 - 2P_1 + P_0) \end{cases}, t = 0 \quad (8)$$

$$\begin{cases} C(1) = P_n \\ \dot{C}(1) = na(P_n - P_{n-1}) \\ \ddot{C}(1) = n(n-1)a^2(P_n - 2P_{n-1} + P_{n-2}) \end{cases}, t = t_{\max} \quad (9)$$

The trajectory beginning and ending points correspond to the first and last control points, respectively. The positions of the first two control points and the last two control points, and the time coefficient can be used to control the speed of the trajectory start and end points, respectively. The positions of the first three control points and the last three control points, and the time coefficient can be used to control the acceleration of the trajectory start and end points, respectively.

To make the foot trajectory smoother and avoid having too many parameters, the foot trajectory of the swing phase is determined by a fifth-order Bézier curve. The foot trajectory is planned in the {H} frame. The coordinates of the foot trajectory control points are given by

$${}^H\mathbf{P}_f = \begin{bmatrix} x_{to} & x_{to} & x_{to} & x_{td} & x_{td} & x_{td} \\ y_{to} & y_{to} & y_{to} & y_{td} & y_{td} & y_{td} \\ z_{to} & z_{to} & z_{to} + h_{foot} & z_{td} + h_{foot} & z_{td} & z_{td} \end{bmatrix} \quad (10)$$

where (x_{to}, y_{to}, z_{to}) is the foot position at take-off, (x_{td}, y_{td}, z_{td}) is the desired foot position at touch-down, and h_{foot} is used to control the maximum ground clearance, which is determined by the obstacle height.

4. Jumping Control Based on SLIP Model

The robot is equivalent to a SLIP model. The virtual spring connects the hip joint to the foot, which is called the virtual leg. The virtual leg can mimic the spring motion by controlling the pitch angle of the knee joint. The jumping process is decoupled into jumping height control, horizontal speed control, and body attitude control using the method proposed by Raibert [11]. In the stance phase, the jumping height and body attitude are controlled by adjusting the ground reaction force. In the flight phase, the horizontal speed of the next jump is changed by controlling the position of the foot.

4.1. Jumping Height Control

In the stance phase, the virtual spring force is adjusted to change the ground reaction force on the foot, in order to regulate the jumping height. The virtual spring force can be expressed as

$$F_{vir} = k_{ss} \left(l_{vir}^d - l_{vir} \right) + k_{ds} \dot{l}_{vir} \quad (11)$$

where k_{ss} is the virtual spring stiffness, k_{ds} is the virtual spring damping, l_{vir}^d is the original length of the virtual spring, and l_{vir} is the virtual spring length.

To account for the system energy loss caused by friction and damping, extra force is added to the spring during the thrust phase of the stance phase. These energy losses vary with speed, and it is difficult to replenish energy manually in real time to maintain the stable jumping of the robot. Cheetahs and greyhounds increase their foot force with the

increase in speed when running and adjust their jumping height by changing the vertical impulse of the stance phase [50]. The logarithmic curve is used to fit the correlation between the peak foot force and the forward speed as the basis for energy supplements. Therefore, the added force is defined as

$$F_{\text{add}} = \alpha \log_a (\dot{x} + 1) \quad (12)$$

where \dot{x} is the forward speed; $a = 2.779$ is the logarithmic base determined by fitting; and α is the amplitude of added force, which is determined by the characteristics of the system, including mass, damping, and friction.

Then, the virtual spring force can be rewritten as

$$F_{\text{vir}} = \begin{cases} k_{\text{ss}}(l_{\text{vir}}^{\text{d}} - l_{\text{vir}}) + k_{\text{ds}}\dot{l}_{\text{vir}}, & \text{at compression phase} \\ k_{\text{ss}}(l_{\text{vir}}^{\text{d}} - l_{\text{vir}}) + k_{\text{ds}}\dot{l}_{\text{vir}} + F_{\text{add}}, & \text{at thrust phase} \end{cases} \quad (13)$$

The virtual spring force calculated with Equation (13) is regarded as the force on the foot. Therefore, according to the geometric relationship, the foot output force can be expressed as

$$\mathbf{F}_{\text{foot}} = \begin{bmatrix} \cos \theta_{\text{vir}} \\ -\sin \theta_{\text{vir}} \sin \theta_1 \\ -\sin \theta_{\text{vir}} \cos \theta_1 \end{bmatrix} F_{\text{vir}} \quad (14)$$

where $\theta_{\text{vir}} = \theta_2 + \arcsin(l_2 \sin \theta_3 / l_{\text{vir}})$ is the virtual leg pitch angle.

The foot Jacobian matrix is used to convert the foot output force to the needed equivalent joint torque.

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F}_{\text{foot}} \quad (15)$$

where $\boldsymbol{\tau} = [\tau_{\text{hr}} \quad \tau_{\text{hp}} \quad \tau_{\text{kp}}]$ and each component represents the torque of hip roll, hip pitch, and knee pitch, respectively.

4.2. Horizontal Speed Control

The horizontal speed can be controlled by adjusting the position of the foot relative to the neutral point, as shown in Figure 2. If the foot is placed on the right side of the neutral point, that is, $x_{f_0} > \dot{x}T_s/2$, part of the kinetic energy is transformed into gravitational potential energy in the stance phase, so the robot's speed decreases, that is, $\dot{x}_i > \dot{x}_{i+1}$. If the foot is placed at the neutral point, that is, $x_{f_0} = \dot{x}T_s/2$, the potential energy at touch-down and take-off is the same, so the robot's speed remains unchanged, that is, $\dot{x}_i = \dot{x}_{i+1}$. If the foot is placed on the left side of the neutral point, that is, $x_{f_0} < \dot{x}T_s/2$, part of the gravitational potential energy is transformed into kinetic energy in the stance phase, so the robot's speed increases, that is, $\dot{x}_i < \dot{x}_{i+1}$.

The acceleration and deceleration control of the robot can be realized by adding an offset $x_{f\Delta}$ on the basis that the foot is placed at the neutral point. The linear estimation of horizontal velocity is used to determine the offset.

$$x_{f\Delta} = k_{\dot{x}}(\dot{x} - \dot{x}_d) \quad (16)$$

where \dot{x}_d is the expected horizontal velocity and $k_{\dot{x}}$ is the speed increment coefficient, which determines how fast the actual speed can reach the desired speed.

To sum up, the expected foot placement at an expected horizontal speed is defined as

$$x_f = \frac{\dot{x}T_s}{2} + k_{\dot{x}}(\dot{x} - \dot{x}_d) \quad (17)$$

According to the inverse kinematics Equation (2), the desired hip joint angle, θ^d , can be obtained. Proportional differential (PD) control is adopted to obtain the hip joint torque to place the foot in the desired position:

$$\tau_i = k_{pf}(\theta_i - \theta_i^d) - k_{df}\dot{\theta}_i, \quad i = \text{hr, hp} \quad (18)$$

where k_{pf} and k_{df} are the position and velocity gain of foot placement, respectively.

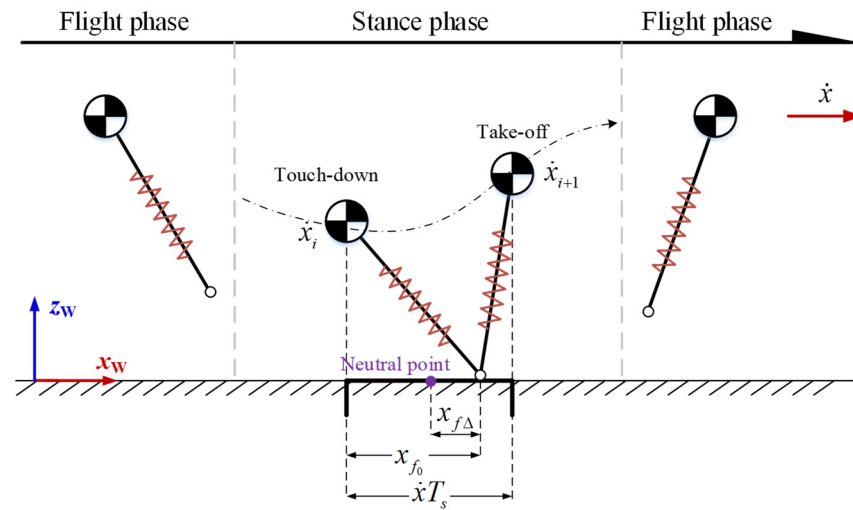


Figure 2. Foot placement and horizontal speed control. x_{f0} is the forward displacement of the foot relative to the center of mass; \dot{x} is the forward horizontal velocity of the center of mass; and T_s is the duration of the stance phase.

4.3. Body Attitude Control

In the stance phase, the body attitude is adjusted by controlling the hip joint torque. The expected body attitude angle is 0. The body attitude adjustment torque is calculated using PD control.

$$\begin{cases} \tau_{hr} = -k_{pb}\theta_{br} - k_{db}\dot{\theta}_{br} \\ \tau_{hp} = -k_{pb}\theta_{bp} - k_{db}\dot{\theta}_{bp} \end{cases} \quad (19)$$

where θ_{br} and θ_{bp} are the body roll and pitch angles, and k_{pb} and k_{db} are the proportional and derivative gains of body attitude, respectively.

4.4. Gravity Compensation

The SLIP model assumes that the mass of the robot is concentrated at the center of mass and ignores the mass of the leg, so that the leg motion does not influence the overall motion of the robot. However, for the leg mechanism in this paper, the gravity of the leg link always has a momentary effect on each joint. Neglecting the mass of the leg would affect the stability of the robot, so gravity compensation is required. The dynamic equation of the robot established using the Lagrange method shows that the moment generated by gravity on each joint is

$$\mathbf{G} = \begin{bmatrix} \frac{1}{2}s_1s_2l_1m_1g + \left(l_1s_1s_2 + \frac{1}{2}l_2s_1s_{23}\right)m_2g \\ -\frac{1}{2}c_1c_2l_1m_1g - \left(l_1c_1c_2 + \frac{1}{2}l_2c_1c_{23}\right)m_2g \\ -\frac{1}{2}c_1c_{23}l_2m_2g \end{bmatrix} \quad (20)$$

The torque to be compensated for each joint is

$$\boldsymbol{\tau} = -\mathbf{G} \quad (21)$$

4.5. Finite-State Machine

To impose the jumping motion engendered from the plan, a finite-state machine (FSM) is established, as shown in Table 2. The robot goes through one stance phase, in which the foot touches the ground, and one flight phase, in which the foot does not touch the ground. The stance phase is split into two sub-phases: compression and thrust. The flight phase is split into two sub-phases: swing and landing. Switching from or to the stance phase occurs when the leg leaves or hits the ground. The transition from or to the stance phase arises when the leg leaves or touches the ground. The FSM executes the specified action in the specified state.

Table 2. The finite-state machine for the jumping cycle.

State	Trigger Event	Action
COMPRESSION	Foot touches ground	a. Leg mimics spring b. Body attitude adjustment c. Gravity compensation
THRUST	Virtual leg extension	a. Leg mimics spring b. Body attitude adjustment c. Energy compensation d. Gravity compensation
SWING	Foot not touching	a. Servo foot trajectory b. Gravity compensation
LANDING	Leg stops swinging	a. Leg mimics spring b. Foot placement c. Gravity compensation

4.6. Controller of One-Legged Hopping Robot

By combining the algorithms described in this section, a stable one-legged-robot jumping motion controller is provided, as shown in Figure 3. The FSM module is responsible for coordinating the execution of actions in different states. The states of the robot are introduced into each module through feedback to form a closed control loop.

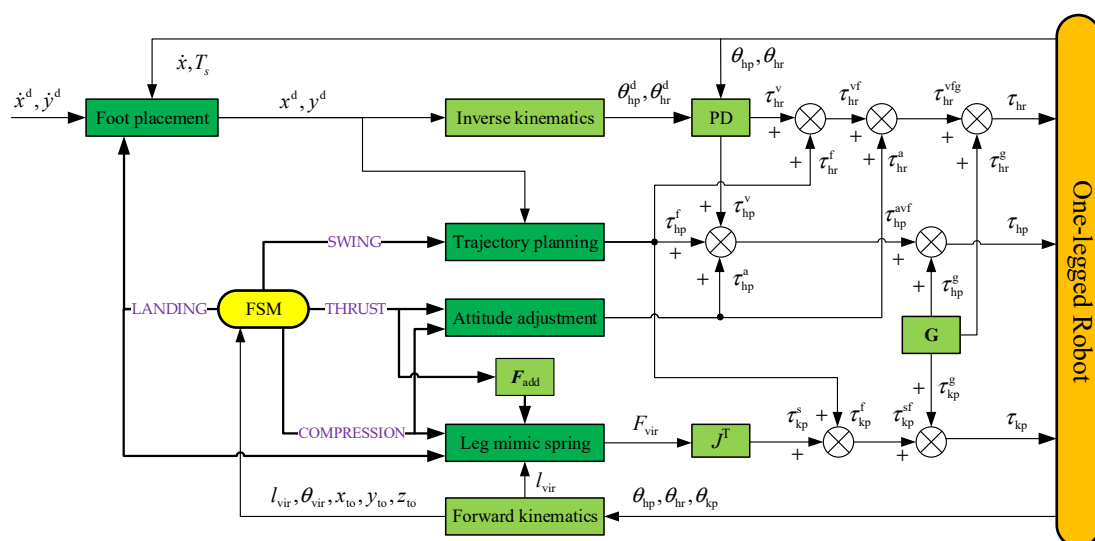


Figure 3. One-legged-robot controller.

5. Jump Simulation Experiment

To examine the performance of the one-legged robot and designed controller, a continuous hopping experiment was carried out. The one-legged-robot model and experimental environment were built in the PyBullet simulation environment. The goal of the robot was to continuously clear obstacles of different heights in the simulation environment. Five obstacles were placed at an interval of 4 m, and the height increased from 0.1 m to 0.5 m. The control parameters set in the simulation environment using parameter tuning are listed in Table 3. Since there was no control method applied to the yaw direction movement, the robot was constrained to moving in a straight line.

In the experiment, the robot jumped forward steadily for a distance of 25 m within 25 s and continuously cleared five walls varying in height from 0.1 to 0.5 m, as shown in Figure 4. The distance between the walls was 4 m, leaving enough distance for the robot to adjust the jumping speed and height to clear the next obstacle. Figure 5 shows the change in body position. The maximum displacement of the body in the vertical direction was 1.4 m, which means that the maximum jump height of the robot reached 0.6 m. In addition, the jumping height could also be enhanced by improving the added foot force.

Table 3. The parameters set in the simulation environment.

Parameter	Symbol	Value
Virtual spring stiffness	k_{ss}	11,000
Virtual spring damping	k_{ds}	60
Original length of the spring	l_{vir}^d	0.675
Amplitude of added force	α	2800
Speed increment coefficient	$k_{\dot{x}}$	0.01
Position gain of foot	k_{pf}	4000
Velocity gain of foot	k_{df}	60
Proportional gain of attitude	k_{pb}	3000
Derivative gain of attitude	k_{db}	80

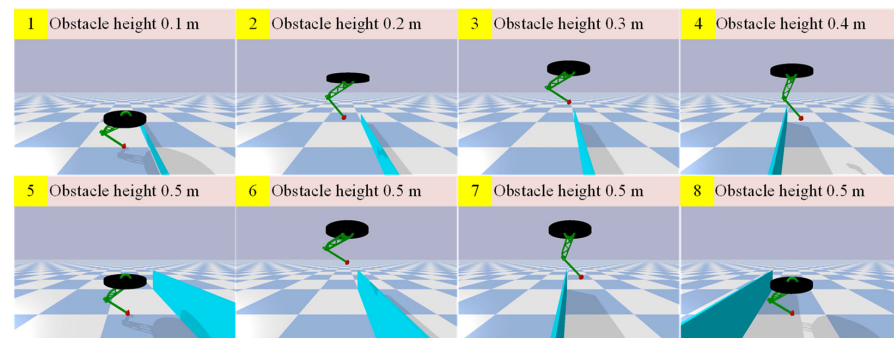


Figure 4. The one-legged robot continuously clears obstacles of different heights.

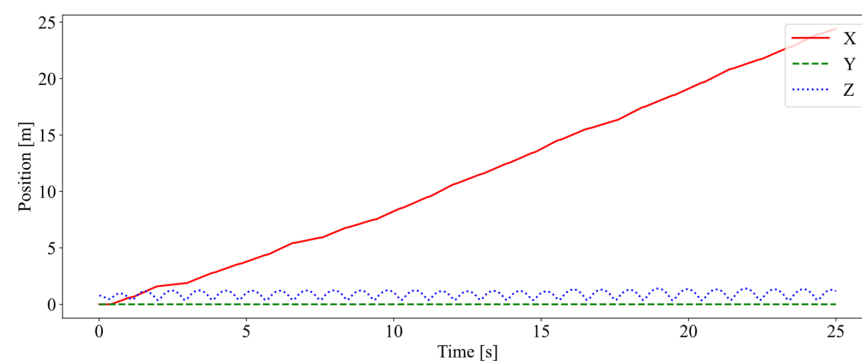


Figure 5. The position of the body as a function of time. X, Y, and Z denote forward, lateral, and vertical directions, respectively.

The velocity of the robot is shown in Figure 6. The robot achieved a forward speed of about 1 m/s at the expected speed of 1.5 m/s. The failure to achieve the desired velocity could be due to the robot tipping over if the foot placement deviated too much from the center of mass. In the meantime, the forward speed fluctuated due to the inaccuracy of foot placement estimation, so the robot constantly adjusted its forward speed during jumping. After entering the stable jump, the vertical speed continued to change steadily as the robot fell and rose.

The desired and actual foot trajectories in the swing phase were nearly consistent, as shown in Figure 7. This indicates that, after entering the swing phase, the controller could quickly plan and follow the foot trajectory according to the position of the foot placement and the ground clearance. Before crossing the obstacle, the robot first lifted its leg, and then set it down after crossing the obstacle, which is similar to humans stepping over steps. In the stance phase, the foot is expected to remain in contact with the ground. There was a period in each jump cycle with the actual foot position remaining 0 in the z-direction and constant in the x-direction, indicating that the foot in the stance phase neither rebounded nor slipped.

Figure 8 shows the three joint torques within 5 s. Jumping is a type of movement in which take-off and touch-down occur alternately. The interaction force between the robot and the external environment changes dramatically as a result of the impact generation and conversion of kinetic and potential energy during the process of touch-down and take-off. As a result, the torque exerted on joints changed periodically throughout the moving process and fluctuated greatly in the stance phase. The torque of the hip pitch was the largest, probably due to the greater pitching motion of the forward jump. As the motion was constrained in the sagittal plane, the torque of the roll hip joint was kept at 0.

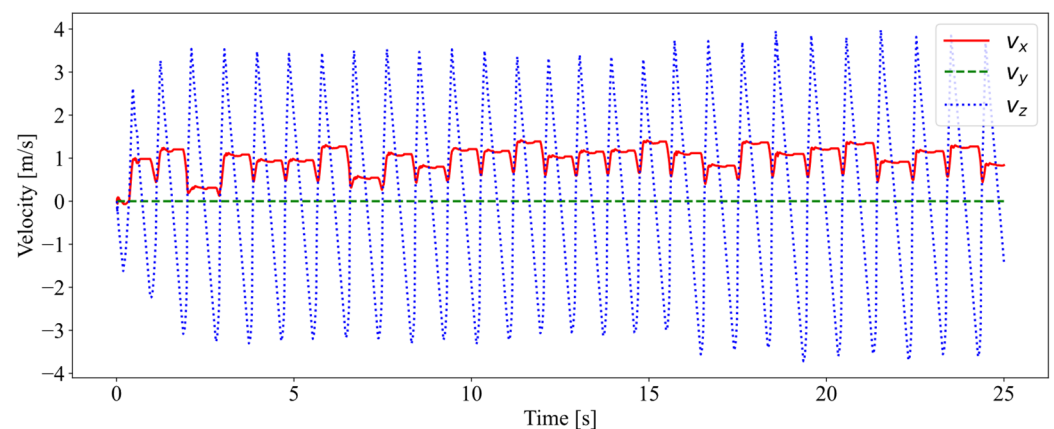


Figure 6. The velocity of the center of mass of the robot.

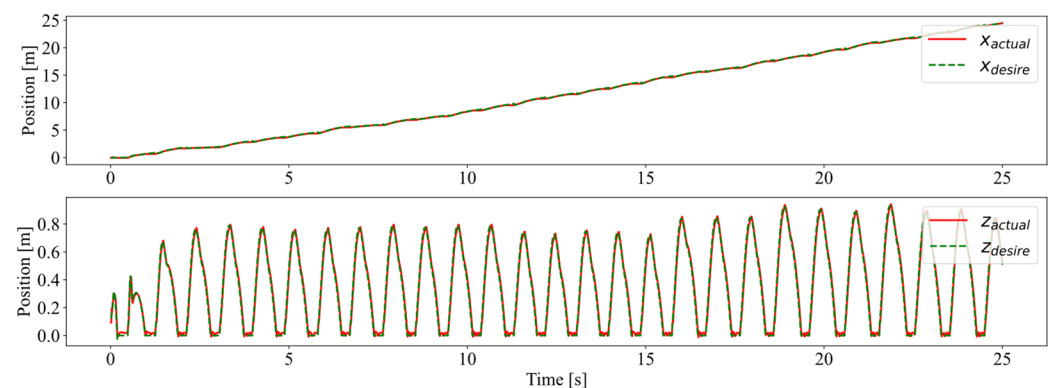


Figure 7. The foot position in the world coordinate system. The solid red and dashed green lines represent the actual and planned foot trajectories, respectively.

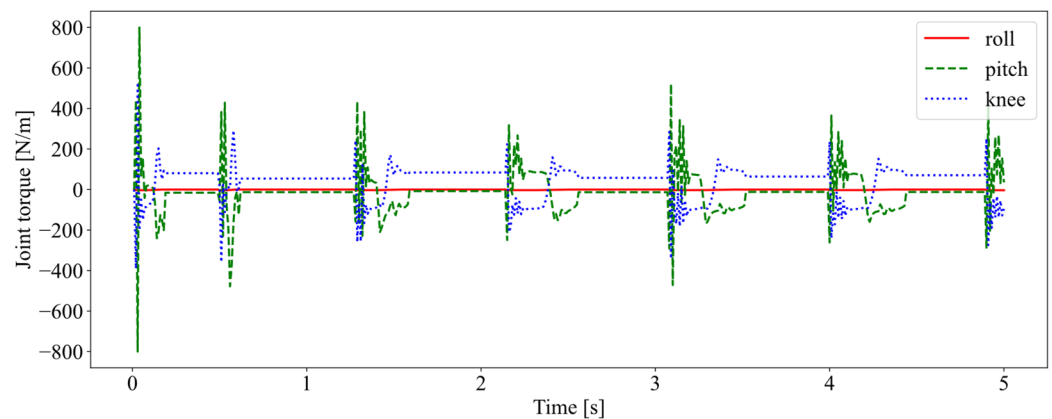


Figure 8. The joint torque in the first 5 s. The solid red, dashed green, and dotted blue lines represent hip roll, hip pitch, and knee pitch torques, respectively.

6. Discussion and Conclusions

In this study, we demonstrate a novel idea in the research area of one-legged-robot control, which applies the Bézier curve to planning foot trajectory and combines the SLIP model to clear obstacles. The foot trajectory of the swing phase is parameterized and generated using a fifth-order Bézier curve. Moreover, the foot force is associated with speed by imitating the running mechanism of the cheetah. Finally, the one-legged robot achieved stable jumping and continuously cleared obstacles of different heights in the simulated experiment, indicating that the control method is effective. In future work, we plan to enlarge the control scheme to achieve omnidirectional motion control. In addition, we plan to transfer the trajectory planning into a nonlinear optimization by applying GA, PSO, and other intelligent optimization methods. In follow-up works, we also plan to train the parameters of the controller based on a reinforcement learning algorithm to achieve a more stable jump.

Author Contributions: Conceptualization, X.Z. and S.H.; methodology, X.Z. and S.H.; software, S.H.; validation, S.H.; writing—original draft preparation, S.H.; writing—review and editing, X.Z. and S.H.; supervision, X.Z.; project administration, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by STI 2030—Major Projects (No. 2021ZD0201403).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the personnel of Bio-inspired Intelligent Robot Lab of Beijing Jiaotong University of China for their technical assistance in the experimental process.

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

References

1. Miki, T.; Lee, J.; Hwangbo, J.; Wellhausen, L.; Koltun, V.; Hutter, M. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **2022**, *62*, eabk2822. [CrossRef] [PubMed]
2. Jeon, S.H.; Kim, S.; Kim, D. Online Optimal Landing Control of the MIT Mini Cheetah. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 178–184.
3. Rathod, N.; Bratta, A.; Focchi, M.; Zanon, M.; Villarreal, O.; Semini, C.; Bemporad, A. Model Predictive Control with Environment Adaptation for Legged Locomotion. *IEEE Access* **2021**, *9*, 145710–145727. [CrossRef]
4. Saputra, A.A.; Botzheim, J.; Ijspeert, A.J.; Kubota, N. Combining Reflexes and External Sensory Information in a Neuromusculoskeletal Model to Control a Quadruped Robot. *IEEE Trans. Cybern.* **2022**, *52*, 7981–7994. [CrossRef] [PubMed]
5. ATLAS. Available online: <https://www.bostondynamics.com/atlas> (accessed on 22 November 2022).

6. Luca, A.D.; Muratore, L.; Tsagarakis, N.G. A Hybrid Primitive-Based Navigation Planner for the Wheeled-Legged Robot CEN-TAURO. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 1–8.
7. Dao, J.; Green, K.; Duan, H.; Fern, A.; Hurst, J. Sim-to-Real Learning for Bipedal Locomotion Under Unsensed Dynamic Loads. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 10449–10455.
8. Huang, Q.; Dong, C.; Yu, Z.; Chen, X.; Li, Q.; Chen, H.; Liu, H. Resistant Compliance Control for Biped Robot Inspired by Humanlike Behavior. *IEEE ASME Trans. Mechatron.* **2022**, *27*, 3463–3473. [\[CrossRef\]](#)
9. Yang, C.; Yuan, K.; Zhu, Q.; Yu, W.; Li, Z. Multi-expert learning of adaptive legged locomotion. *Sci. Robot.* **2022**, *5*, eabb2174. [\[CrossRef\]](#)
10. Chi, W.; Jiang, X.; Zheng, Y. A Linearization of Centroidal Dynamics for the Model-Predictive Control of Quadruped Robots. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4656–4663.
11. Raibert, M.H. *Legged Robots That Balance*; MIT Press: Cambridge, MA, USA, 1986.
12. Weng, Y.; Chen, X.; Yu, Z.; Qi, H.; Ma, X.; Zhu, M.; Huang, Q. Leg Mechanism Design of a Jumping Robot with Variable Reduction Ratio Joint. In Proceedings of the 2022 International Conference on Intelligent Robotics and Applications, Harbin, China, 1–3 August 2022; pp. 250–261.
13. Yim, J.K.; Fearing, R.S. Precision Jumping Limits from Flight-phase Control in Salto -1P. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 2229–2236.
14. He, Z.; Meng, F.; Chen, X.; Yu, Z.; Fan, X.; Sato, R.; Huang, Q. Controllable height hopping of a parallel legged robot. *Appl. Sci.* **2021**, *11*, 1421. [\[CrossRef\]](#)
15. Chen, Z.; Jin, B.; Zhu, S.; Pang, Y.; Chen, G. Dynamic Hopping Height Control of Single-Legged Hopping Robot. *Wearable Sens. Robot.* **2017**, *399*, 365–382.
16. Kalouche, S. GOAT: A legged robot with 3D agility and virtual compliance. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4110–4117.
17. Ugurlu, B.; Kawamura, A. ZMP-based online jumping pattern generation for a one-legged robot. *IEEE Trans. Ind. Electron.* **2010**, *57*, 1701–1709. [\[CrossRef\]](#)
18. Jiang, X.; Chen, X.; Yu, Z.; Zhang, W.; Meng, L.; Huang, Q. Motion planning for bipedal robot to perform jump maneuver. *Appl. Sci.* **2018**, *8*, 139. [\[CrossRef\]](#)
19. Park, H.W.; Wensing, P.M.; Kim, S. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In Proceedings of the 2015 Robotics: Science and Systems Conference (RSS), Rome, Italy, 13–17 July 2015; pp. 1–9.
20. Nguyen, C.; Bao, L.; Nguyen, Q. Continuous Jumping for Legged Robots on Stepping Stones via Trajectory Optimization and Model Predictive Control. *arXiv* **2022**, arXiv:2204.01147.
21. Chignoli, M.; Kim, D.; Stanger-Jones, E.; Kim, S. The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors. In Proceedings of the 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), Munich, Germany, 19–21 July 2021; pp. 1–8.
22. Chignoli, M.; Morozov, S.; Kim, S. Rapid and reliable quadruped motion planning with omnidirectional jumping. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 6621–6627.
23. Chignoli, M. Trajectory Optimization for Dynamic Aerial Motions of Legged Robots. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2021.
24. Song, Z.; Yue, L.; Sun, G.; Ling, Y.; Wei, H.; Gui, L.; Liu, Y.H. An Optimal Motion Planning Framework for Quadruped Jumping. *arXiv* **2022**, arXiv:2207.12002.
25. Nguyen, C.; Nguyen, Q. Contact-timing and Trajectory Optimization for 3D Jumping on Quadruped Robots. *arXiv* **2022**, arXiv:2110.06764.
26. Nguyen, Q.; Powell, M.J.; Katz, B.; Carlo, J.D.; Kim, S. Optimized Jumping on the MIT Cheetah 3 Robot. In Proceedings of the 2019 IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 7448–7454.
27. Bellegarda, G.; Nguyen, Q. Robust quadruped jumping via deep reinforcement learning. *arXiv* **2020**, arXiv:2011.07089.
28. Campanaro, L.; Gangapurwala, S.; Martini, D.D.; Merkt, W.; Havoutis, I. CPG-ACTOR: Reinforcement Learning for Central Pattern Generators. *arXiv* **2021**, arXiv:2102.12891.
29. Mo, X.; Ge, W.; Romano, D.; Donati, E.; Stefanini, C. Modelling jumping in locusta migratoria and the influence of substrate roughness. *Entomol. Gen.* **2019**, *38*, 317–332. [\[CrossRef\]](#)
30. Fercher, C. The biomechanics of movement of horses engaged in jumping over different obstacles in competition and training. *J. Equine Vet. Sci.* **2017**, *49*, 69–80. [\[CrossRef\]](#)
31. Alexander, R.M. Leg design and jumping technique for humans, other vertebrates and insects. *Philos. Trans. R. Soc. London. Ser. B Biol. Sci.* **1995**, *347*, 235–248.
32. Gabriel, J.M. The effect of animal design on jumping performance. *J. Zool.* **1984**, *204*, 533–539. [\[CrossRef\]](#)
33. Alexander, R.M. The mechanics of jumping by a dog (*Canis familiaris*). *J. Zool.* **1974**, *173*, 549–573. [\[CrossRef\]](#)
34. Alexander, R.M.; Vernon, A. The mechanics of hopping by kangaroos (*Macropodidae*). *J. Zool.* **1975**, *177*, 265–303. [\[CrossRef\]](#)

35. Burrows, M. Jumping performance of froghopper insects. *J. Exp. Biol.* **2006**, *209*, 4607–4621. [[CrossRef](#)] [[PubMed](#)]
36. Mo, X.; Ge, W.; Miraglia, M.; Inglese, F.; Zhao, D.; Stefanini, C.; Romano, D. Jumping locomotion strategies: From animals to bioinspired robots. *Appl. Sci.* **2020**, *10*, 8607. [[CrossRef](#)]
37. Pfau, T.; Rivaz, A.; Brighton, S.; Weller, R. Kinetics of jump landing in agility dogs. *Vet. J.* **2011**, *190*, 278–283. [[CrossRef](#)] [[PubMed](#)]
38. Blickhan, R. The spring-mass model for running and hopping. *J. Biomech.* **1989**, *22*, 1217–1227. [[CrossRef](#)] [[PubMed](#)]
39. Mo, X.; Romano, D.; Miraglia, M.; Ge, W.; Stefanini, C. Effect of substrates' compliance on the jumping mechanism of locusta migratoria. *Front. Bioeng. Biotechnol.* **2020**, *8*, 661. [[CrossRef](#)]
40. Farley, C.T.; Gonzalez, O. Leg stiffness and stride frequency in human running. *J. Biomech.* **1996**, *29*, 181–186. [[CrossRef](#)]
41. Ferris, D.P.; Louie, M.; Farley, C.T. Running in the real world: Adjusting leg stiffness for different surfaces. *Proc. R. Soc. Lond.* **1998**, *265*, 989–994. [[CrossRef](#)]
42. Kim, S.; Park, S. Leg stiffness increases with speed to modulate gait frequency and propulsion energy. *J. Biomech.* **2011**, *44*, 1253–1258. [[CrossRef](#)]
43. Guo, W.; Cai, C.; Li, M.; Zha, F.; Wang, P.; Jiang, Z. Estimation of leg stiffness using an approximation to the planar spring–mass system in high-speed running. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881419890713. [[CrossRef](#)]
44. Yin, P.; Wang, P.; Li, M.; Sun, L. A Novel Control Strategy for Quadruped Robot Walking Over Irregular Terrain. In Proceedings of the 2011 IEEE Conference on Robotics, Automation and Mechatronics (RAM), Qingdao, China, 17–19 September 2011; pp. 184–189.
45. Kim, K.Y.; Park, J.H. Ellipse-based leg-trajectory generation for galloping quadruped robots. *J. Mech. Sci. Technol.* **2009**, *22*, 2099–2106. [[CrossRef](#)]
46. Sakakibara, Y.; Kan, K.; Hosoda, Y.; Hattori, M.; Fujie, M. Foot Trajectory for a Quadruped Walking Machine. In Proceedings of the 1990 IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications, Ibaraki, Japan, 3–6 July 1990; pp. 315–322.
47. Rong, X.; Li, Y.; Ruan, J.; Li, B. Design and simulation for a hydraulic actuated quad-ruped robot. *J. Mech. Sci. Technol.* **2012**, *26*, 1171–1177. [[CrossRef](#)]
48. Yuan, L.; Zhang, Z.; Ouyang, R. Trajectory Planning of Quadruped Robot Based on the Principle of Optimal Power. *Int. J. Mech. Res.* **2016**, *5*, 138–147. [[CrossRef](#)]
49. Hyun, D.J.; Seok, S.; Lee, J.; Kim, S. High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *Int. J. Robot. Res.* **2014**, *33*, 1417–1445. [[CrossRef](#)]
50. Hudson, P.E.; Corr, S.A.; Wilson, A.M. High speed galloping in the cheetah (*Acinonyx jubatus*) and the racing greyhound (*Canis familiaris*): Spatio temporal and kinetic characteristics. *J. Exp. Biol.* **2012**, *215*, 2425–2434. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.