Vol. 59, No. 9
May 2023

machine Mechanical Engineering
Engineering Journal of Mechanical
Engineering Journal
Journal of Mechanical Engineering

Vol. 59 No. 9
May 2023

DOI: 10.3901/JME.2023.09.051

Wheel-leg self-balancing robot based on a nonlinear spring model
A Study of Jumping Algorithms*

Gao Jing Song[1] Kim Hong Chul[1] Zhu Yan He[1] Gao Liang[1] Lv Hong Ya [2]
Zhao Jie[1] Cai Hegao [1]

(1. State Key Laboratory of Robotics Technology and Systems, Harbin Institute of Technology, Harbin 150001, China; 2. Shanghai Aerospace Equipment Manufacturing Co. 150001; 2. Shanghai Aerospace Equipment Manufacturing General Factory Co. 200000)

Abstract: Wheel-legged self-balancing robots combine the high speed and efficiency of wheeled type and the ground adaptability of footed type, and can jump over obstacles when facing unstructured terrain. According to the leg degrees of freedom, they can be categorized into single-degree-of-freedom type and two-degree-of-freedom type, among which the single-degree-of-freedom type wheel-legged self-balancing robots are simpler in structure, lighter in mass, and less difficult to control. However, in the jump trajectory planning problem, on the one hand, the single-degree-of-freedom leg structure requires higher force from the hip joint, and the maximum height of obstacle crossing can be limited by using the trajectory planning method of the linear spring model with two mass blocks; on the other hand, the overall center of mass of the robot will be displaced in the x-direction during the height adjustment process, which will have an impact on the accuracy and stability of the jump. In order to study the jumping problem of a single-degree-of-freedom wheel-legged self-balancing robot, a wheel control algorithm based on the airborne dynamics model is firstly proposed, so that the body's pitching attitude can be controlled during the jumping process, and the stability of the jumping is thus ensured. After that, a jump trajectory planning method based on the nonlinear spring model of double mass block is proposed, which has the advantages of more flexible trajectory planning and lower requirements for hip joint force than the planning method based on the linear spring model; then, the control effect of the wheel part on the pitch angle of the fuselage during the process of vacating is further utilized, and a method of in-situ jumping is designed, so that the robot can reach the same jumping distance with shorter jumping time and starting distance; finally, the control algorithm of wheel part is established, so that the body pitching attitude is always controlled during the process of jumping, thus ensuring the stability of the jumping. Finally, a three-dimensional simplified model of a single-degree-of-freedom wheel-legged self-balancing robot and its kinematics, single-leg static mechanics, and airborne dynamics are established, and the feasibility of the trajectory planning and tracking algorithms is verified through Simulink-Adams co-simulation.

Keywords: wheel-legged self-balancing robot; trajectory planning; jump control; nonlinear spring C.I.S.: TP242

Research on Jumping Algorithm of Wheel-legged Self-balancing Robot

Based on Nonlinear Spring Model

GAO Jingsong[1] JIN Hongzhe[1] ZHU Yanhe[1] GAO Liang[1] LÜ Hongya[2] ZHAO Jie[1] CAI Hegao[1]

(1. State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001;
2. Shanghai Aerospace Equipments Manufacturer Co. Ltd, Shanghai 200000)

Abstract： The wheel-legged self-balancing robot has both the high speed and efficiency of the wheeled type and the ground
According to the degree of freedom of the legs, it can be divided into single-degree-of-freedom type and two-degree-of-freedom type. According to the degree of freedom of the legs, it can be divided into single-degree-of-freedom type and two-degree-of-freedom type. Among them, the single-degree-of-freedom type has a simpler structure, lighter weight and less difficult to control. Among them, the single-degree-of-freedom type has a simpler structure, lighter weight and less difficult to control. However, on the problem of jump trajectory planning, on the one hand, the leg structure has a higher demand for the output of the hip. higher demand for the output of the hip joint, which limits the maximum obstacle height that can be achieved by using the two-mass linear spring model trajectory planning method; On the other hand, the overall center of mass will move in the x-direction during the height adjustment process, which will affect the accuracy and stability of jumping. A research on the jumping problem of a single-degree-of-freedom wheel-legged self-balancing robot is carried out. carried out.

* Supported by the National Natural Science Foundation of China (92048301) and the Science and Technology Innovation 2030 - "Brain Science and Brain-like Research" Major (2021ZD0201403). 20220519 Received the first draft, 20221108 Received the revised draft.

in order to ensure the stability of jumping, a wheel control algorithm is proposed to control the pitch attitude of the fuselage during the whole jumping After that, a trajectory planning method based on a two-mass nonlinear spring model is proposed and proved to be more flexible and powerful comparing with the planning method based on the two-mass linear-spring model. After that, a trajectory planning method based on a two-mass nonlinear spring model is proposed and proved to be more flexible and powerful comparing with the planning method based on the two-mass linear-spring model. Then, a method of long jump in situ is designed by using the wheel control algorithm furtherly, so that the robot can achieve the same jumping distance in a shorter takeoff time and distance. The three-dimensional model of the robot is established, as well as its kinematics, single-leg statics and aerial dynamics models. The jump trajectory planning and tracking algorithms proposed in this paper is realized in the robot. The jump trajectory planning and tracking algorithms proposed in this paper is realized in the Simulink-Adams simulation and proved to be feasible.

Key words: wheel-legged self-balancing robot; trajectory planning; jumping algorithm; nonlinear spring

0 Preamble

All-terrain mobile robots have both indoor and outdoor operation capabilities[1] , which are widely used in industrial inspection, field exploration and other scenarios[2-3] . Among them, wheel-legged self-balancing robots (hereinafter referred to as wheel-legged robots), as a kind of composite mobility mode robots with both wheeled high-speed and high efficiency and footed ground adaptation, have rapidly become one of the research hotspots at home and abroad in recent years. Jumping is the main way for wheel-legged robots to cope with unstructured terrain, so the jumping ability of the robot has become one of the most important indexes to measure its environmental adaptability, and the advanced wheel-legged robots at home and abroad have good jumping ability, for example, the Handle[4] robot of Boston Dynamics, the Ascento robot of ETH Zurich[5-6] , and the Ollie robot of Tencent[7] , and so on. Early research on jumping algorithms for wheel-legged robots mainly referred to single-legged jumping robots, and the most mainstream trajectory planning methods are center-of-mass motion-based planning methods and SLIP model-based planning methods[8] . The former equates the mass of the robot to a prime point, and realizes the jump trajectory planning based on the analysis of the kinematics and dynamics of this prime point[9] , for example, the QRIO running robot developed by Sony[10] and the single-legged jumping robot designed by C.W. Chen[11] . The SLIP model, on the other hand, equates the robot leg to a massless spring, concentrates the mass at the tip of the spring, and characterizes the jumping motion by means of a spring-loaded system. FULL[12] and DICKINSON[13] are based on the SLIP model.

The FRANK jumping robot and the HYL single-leg prototype have been developed to realize the stable jumping of the robot. MATHIS[14] proposes a two-mass block spring model based on the SLIP model, which takes into account the effect of the lower center of mass in the process of airborne and landing, so as to make the established dynamics model more accurate. Besides, there are jump trajectory planning methods based on optimal control[15] and stability[16] .

Based on the above research results, Fan Yixiao[17] carried out a study on the jumping mechanism of articulated legged robot based on the dual mass block spring inverted pendulum model and a study on the landing contact force control strategy based on the optimal approach speed, which can reduce the impact force of the robot on the ground by 70.71%, and the error of the height of the wheels off the ground can be controlled within 3%; Li[18] carried out a study on the jump height, landing buffer, attitude, and balance control of robot based on the TMS-DIP model, respectively. Li investigated the jump height, landing cushion, attitude and balance control of the robot based on the TMS-DIP model. The results show that compared with the spring-loaded inverted pendulum model, the landing impact force of the TMS-DIP model is much lower than that of the spring-loaded inverted pendulum model.

Yu Hao-Yang[19] simplified the wheel-legged robot into a dual-mass spring-loaded inverted pendulum (DM-SLIP) model and planned the trajectories of jumping phases based on this model to get the maximum height of obstacle-crossing that the model can reach, and mapped the center-of-mass trajectories into the joint space through the quadratic planning method, so that the WLR-4P robot leaped onto a 0.4 m platform and landed smoothly; CHEN proposed a wheel-type spring-loaded inverted pendulum (W-SLIP) model and planned the center-of-mass trajectories of jumping phases based on the quadratic planning method. The WLR-4P robot jumped onto a 0.4 m platform and landed smoothly; CHEN[20] proposed a wheeled spring-loaded inverted pendulum (W-SLIP) model, based on the quadratic programming method, the center of mass trajectories of the jumping phases were planned, and the composite controller of the standing phase based on the perturbation observer and the whole-body controller of the vacating phase in the standard task space were established, and the performance of the whole framework was verified by V-REP simulation.

Existing research on the jump control of wheel-legged robots mainly focuses on robots with a two-degree-of-freedom leg structure, and based on the two-mass linear spring model (hereinafter referred to as the linear spring model), the virtual model trajectory is mapped to the joint space through the dynamics model to realize the jump function. The robot with single-degree-of-freedom leg structure can only achieve z-direction trajectory tracking by controlling the hip joint. In order to reduce the x-direction displacement of the upper center of mass during the jumping process so as to ensure that the robot does not overturn in the process of vacating, it is necessary to try to reduce the mass of the leg and optimize the position of the body counterweight; at the same time, the single-degree-of-freedom wheeled-legged robots have a greater demand for the output force of the hip joint, which restricts the maximum height of the robot. The maximum height of the robot over obstacles is limited. In addition, the jumping action of wheel-legged robots often adopts the decoupling of the wheel and the leg, so that the robot has enough moving speed at the time of jumping through the wheel, but the robot accelerates to the specified speed needs enough time and jumping distance, so when the robot needs to jump immediately or does not have enough accelerating distance, its jumping function will be greatly limited. In order to solve the above problems, the innovative work of this paper is reflected in the following aspects: firstly, a wheel action-based pitch angle control algorithm for the body of the vacating phase is proposed, which compensates for the x-direction displacement and the initial velocity introduced in the process of jumping, and ensures the stability of the robot's jumping; and then a jump trajectory planning method based on the dual-mass-block-nonlinear-spring model (hereinafter referred to as the nonlinear spring model) is proposed, which is better than that of the planning method based on the linear spring model, and the method is better than that of the planning method based on the linear spring model. Compared with the planning method based on linear spring model, this method can realize a larger jump height under the same hip joint force, and more constraints can be imposed on the jump trajectory to improve the robot's obstacle-crossing ability and controllability. Finally, an in-situ long jump algorithm is designed to shorten the jumping time required for the robot to reach the same jumping distance at the same jumping height.

time and jumping distance, which improves the robot's ability to jump long distances.

The rest of this paper is organized as follows: Part I describes the control architecture of the robot jumping algorithm and the robot platform used, and establishes the required kinematics and dynamics models; Part II describes in detail the trajectory planning, solving, and mapping methods based on the nonlinear spring model, as well as the trajectory tracking and control algorithm; Part III demonstrates the simulation results of the in-situ height adjustment, the jumping while traveling, and the in-situ long jump; Part IV summarizes the advantages, disadvantages, and control algorithm described in this paper. Part IV summarizes the advantages and disadvantages of the trajectory planning and control algorithms described in this paper. Remark 1: Firstly, by reviewing the literature [21], we determine the trajectory control algorithm that contains a one-time term and cubic term of the nonlinear spring expression. A reasonable design of the spring's coefficient of strength under the same conditions of compression and pressure can store more elastic potential energy and thus achieve a greater jump height, as shown in equation (1)

$$\begin{cases} F_1 = k\Delta l \\ F = a\Delta l^3 + b\Delta l \\ E_1 = \dfrac{1}{2}k\Delta l^2 \end{cases} \tag{1}$$

$$\begin{cases} E = \dfrac{1}{4}a\Delta l^4 + \dfrac{1}{2}b\Delta l^2 \end{cases}$$

Assuming that the spring compression is the same as Δl, the coefficients of strength of the linear and nonlinear springs are k, a, and b, respectively, and the spring support forces are F_1, F_2, the stored elastic potential energy is E_1, E_2

The support of the spring is equal,

then one can find the

$$k = a\Delta l^2 + b \tag{2}$$

At this time the difference between the elastic potential energy of the two springs is

$$\Delta E = E_1 - E_2 = a\Delta l^4 / 4 \tag{3}$$

The jump phase provides the robot with the kinetic energy needed for jumping; the body phase enables the robot to meet the obstacle crossing requirements by planning the trajectory of the upper and lower centers of mass; the landing preparation phase zeroes the falling speed of the wheel part at the moment of landing to reduce the impact between the wheel part and the ground; the landing phase completes the robot's cushioning action and adjusts the robot's attitude to the initial jumping state so as to realize the successive jumps, as shown in Fig. 1.

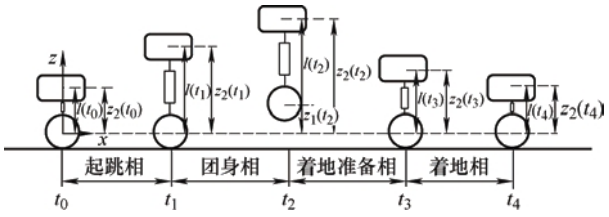


Fig. 1 Robot jumping stage division

Based on the division of the jump stages in Fig. 1, define the stages of the The switching moments are $t \sim t$ with the center of the wheel at time t as the coordinate.

0 4 0

origin, the parameters for the robot and the virtual model are defined in Table 1 Shown.

Table 1 Definitions of jump-related parameters

Name Symbol/Unit	Name Symbol/Unit
Lower mass block traverse distance $x_1(t)$/m Lower mass block rise distance $z_1(t)$/m Upper mass block traverse distance $x_3(t)$/m Upper mass block rise distance $z_2(t)$/m Integral center of mass traverse distance $x_3(t)$/m Integral center of mass rise distance $z_3(t)$/m Jump phase time period t_{01}/s Group Body Phase Time Period t_{12}/s Landing Preparation Phase	Support force to compensate for gravity F_m/N Total support force between mass blocks F/N Fuselage pitch angle target value θ_t /rad Jump target pitch angle deviation θ_b /rad Fuselage pitch angle set value θ_s /rad Actual Fuselage Pitch Angle θ /rad Travel speed setting value v /(m·s)$^{-1}$ Actual value of moving

The two bullets were made to

From equation (3), it can be seen that the use of the nonlinear spring model containing only primary and cubic terms not only ensures the monotonicity of its outgoing force, which is convenient for the leg to realize the jumping function through trajectory tracking, but also in the case of the same amount of compression and pressure, if the nonlinear spring stiffness coefficient $a<0$, the spring will store more elastic potential energy, and a larger jumping height can be achieved. This property of nonlinear springs can be utilized to solve the problem of limited maximum height of obstacle crossing for single-degree-of-freedom wheel-legged robots.

1 Problem statement

In this section, the jump trajectory planning and control method proposed in this paper is firstly briefly introduced, and the parameters involved in the trajectory planning process are explained. After that, the single-degree-of-freedom wheel-legged robot platform used in this paper is introduced and its kinematics, single-leg dynamics, and vacillation dynamics are modeled.

1.1 Virtual model jump trajectory planning and control method

In planning the robot jumping trajectory, the jumping process is divided into four phases: the jumping phase, the regimentation phase, the landing preparation phase and the landing phase.

Aiming at the single-degree-of-freedom leg structure of the wheel-legged robot, which can only realize the z-direction trajectory planning, the wheel control firstly realizes that the body pitch angle is controllable throughout the whole process, so as to ensure the stability of jumping. The wheel landing is based on the two-wheeled inverted pendulum model[22] with variable relative heights of the fuselage and the center of mass of the wheel, and adopts the three-loop parallel PID control algorithm based on gain scheduling, and the upright loop, steering loop, and walking loop PIDs realize the control of the robot's pitch angle, yaw angle, and travel speed, respectively, and the PID parameters of the upright loop and the speed loop change with the angle of the hip joints; when the wheel part is vacating, it is based on a model of the vacating dynamics, and adopts the upright-loop PID control algorithm. When the wheel is vacating, based on the vacating dynamics model, the upright loop PID control algorithm is used to ensure that the body pitch angle changes according to the established target value, see equation (4).

$$\tau_w(t_{01}) = \tau_w(t_{34}) = K_{p\theta 1}(\theta_4)[\theta \boxtimes \theta_s(\theta_4)] + K_{d\theta 1}(\theta_4)\dot{\theta} \pm [K_{p\delta}(\delta \boxtimes \delta_s) + K_{d\delta}\dot{\delta}] + K_{pv}(\theta_4)(v \boxtimes v_s) + K_{iv}(\theta_4)\int_0^t (v \boxtimes v_s)dt,\ t \in [t_0, t_1] \cup [t_3, t_4]$$

$$\tau_w(t_{13}) = K_{p\theta 2}(\theta_4)[\theta \boxtimes \theta_s(\theta_4)] + K_{d\theta 2}(\theta_4)\dot{\theta},\ t \in (t_1, t_3) \quad (4)$$

The critical proportionality method is used to self-tune the PID parameters[23] , and the optimal parameters of the PID controller are derived from the empirical equations in Table 2.

Table 2 Empirical equations for parameterization of Z-N critical scale method

Parameter	K_p	T_I	T_D
P-Type Controller	0.5 K_u	None	None
PI Type Controller	0.45 K_u	0.83 T_u	not
PID type controller	0.6 K_u	0.5 T_u	have
			0.125 T_u

For leg control, due to the establishment of a wheel-legged robot whole machine dynamics

In order to realize jumping by planning the trajectory of the virtual model and mapping it to the joint space of the robot, see Fig. 2. Firstly, in order to fully consider the trajectory of the wheel part of the robot during the process of vacating, the virtual model is used to realize the jumping. Trajectory planning and the impact force between the wheel and the ground at the moment of landing, and in order to realize a higher jump height with the same hip moment, a nonlinear spring model is used as the object of trajectory planning in this paper. Since the overall center of mass of the robot's body and legs is basically kept directly above the wheels when the robot is self-balancing, the wheels are considered as the lower mass block, and the rest of the mass is considered as the upper mass block. Based on the above virtual model, the upper and lower center of mass trajectories can be planned to obtain the height difference between the upper and lower centers of mass and their changing speeds as well as the center of mass.

Curve of the support force between the volume blocks with respect to time. This is followed by a kinematic modeling

The type maps the hip joint angle to the virtual model space in the virtual model

Running the trajectory tracking algorithm in space to get the required support between the dual mass blocks

holding force, which is mapped back to the hip-gate by a single-leg kinetic modeling

The section space realizes the jumping action.

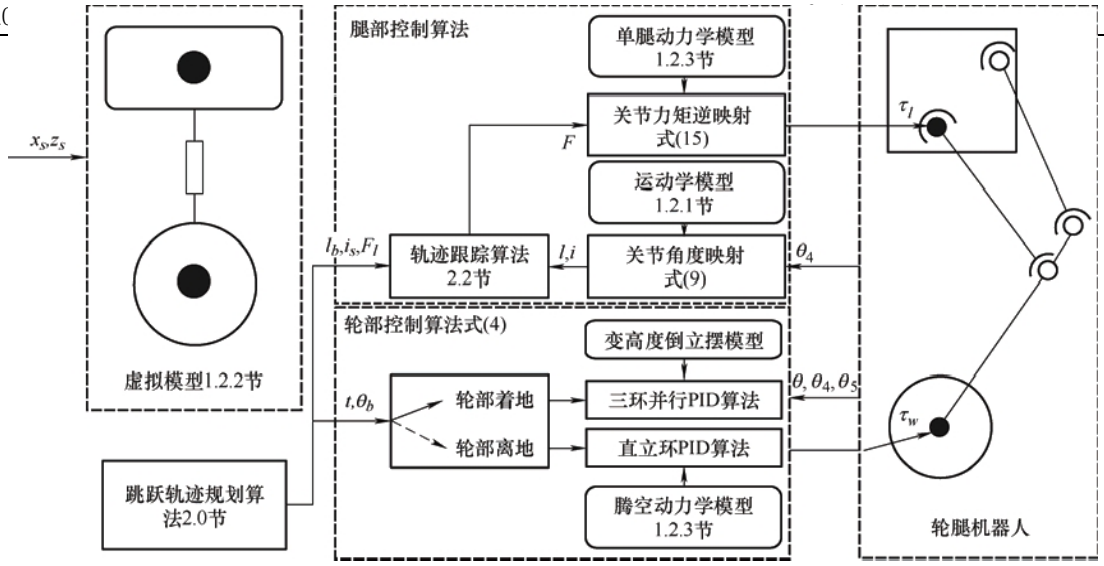


Fig. 2 Robot jump control architecture

1.2 model description

The wheel-legged robot platform used in this paper refers to the Ascento[2] four-link leg configuration, which realizes a nearly straight motion trajectory of the end of the foot through the design of parameters such as the length of each link and its angle with the fuselage. The overall center of mass of the fuselage and legs is located near the line between the wheel and hip centers, which ensures that the target pitch angle of the robot varies within a small range during the height adjustment process, as shown in Fig. 3.

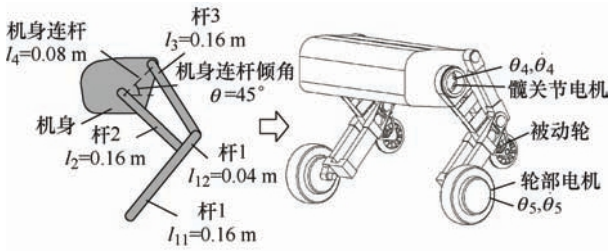


Fig. 3 Single-degree-of-freedom wheel-legged robot platform

The robot dynamics parameters determined from the 3D model are shown in Table 3.

Table 3 Wheel-legged robot dynamics parameters

variant	Symbol/Unit	variant	Symbol/Unit
wheel radius	R /m	Rod 2 Center of mass length	L_{m2} /m
Wheel mass	m_w /kg		L_{m3} /m
Lower Mass Block Mass	m_1 /kg	Rod 3 Center of mass length Body center of mass length	L_{m4} /m
			β_{m1} /rad
Rod 1 mass	m_{L1} /kg		β_{m2} /rad
Rod 2 mass	m_{L2} /kg		β_{m3} /rad
Rod 3 Quality	m_{L3} /kg		β_{m4} /rad
body mass	m_b /kg	Rod 1 Center of mass eccentricity	I_2 /(kgim^2)
Upper mass block mass	m_2 /kg		I_1 /(kgim^2)
Rod 1 Center of mass length	L_{m1} /m	Rod 2 Mass eccentricity	
		Rod 3 Center of mass eccentricity	
		Body center of mass eccentricity	
		Inertia of the upper mass block around the wheel axis	
		Moment of inertia of the wheel around the wheel axis	

1.2.1 Robot kinematics modeling

Robot kinematics modeling with wheel center as coordinate origin as shown in Fig. 4. The hip angle is solved by kinematic modeling, Angular velocity and distance between mass blocks above and below the virtual model and its speed of change

The relationship between degrees. θ_2 ,θ_3 ,θ_4 can be solved by solving for the interior angles of the quadrilateral, and the acceleration relationship requires a system of equations using the linkage vector relationship

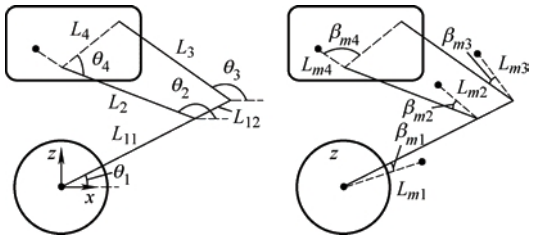


Fig. 4 Robot kinematics model

$$\begin{aligned} & L_2\sin(\theta_2 - \theta_1)\dot{\theta}_1 - L_4\sin(\theta_2 + \theta_4 - \theta)\dot{\theta}_1 \\ & = L_3\sin(\theta_3 - \theta)\dot{\theta}_1 \\ & L_2\cos(\theta_2 - \theta_1)\dot{\theta}_1 - L_4\cos(\theta_2 + \theta_4 - \theta)\dot{\theta}_1 \end{aligned}$$

$$= L_{12} + L_3\cos(\theta_3 - \theta)\dot{\theta}_1 \quad (5)$$

The partial derivatives of Eq. (5) for θ_2, θ_3, and θ_4 can be obtained as $\dot{\theta}$, $\dot{\theta}$, and $\dot{\theta}$, respectively.

The relationship between $\dot{\theta}$

$$\begin{cases} \dfrac{\partial\dot{\theta}_2}{\partial\dot{\theta}_4} = \dfrac{-L_4\sin(\theta_3 - \theta_4)}{L_2\sin(\theta_2 - \theta_3 + \theta_4) - L_3\sin(\theta_2 - \theta_3)} \\ \dfrac{\partial\dot{\theta}_3}{\partial\dot{\theta}_4} = \dfrac{L_2L_4\sin\theta_4}{L_2L_3\sin(\theta_2 - \theta_3) - L_3L_4\sin(\theta_2 - \theta_3 + \theta_4)} \end{cases} \quad (6)$$

Establish the center of mass

$$\begin{cases} l(\theta_4) = \sqrt{x_{ms}^2(\theta_4) + z_{ms}^2(\theta_4)} \\ \dot{l}(\theta_4,\dot{\theta}_4) = \dfrac{x_{ms}\dot{x}_{ms} + z_{ms}\dot{z}_{ms}}{l(\theta_4)} \end{cases} \quad (9)$$

1.2.2 Virtual modeling of nonlinear springs with two mass blocks

The hub motor is considered as the lower mass block and the rest of the mass is considered as the upper mass block to establish a nonlinear spring dynamics model for the two-mass block is shown in Fig. 5.

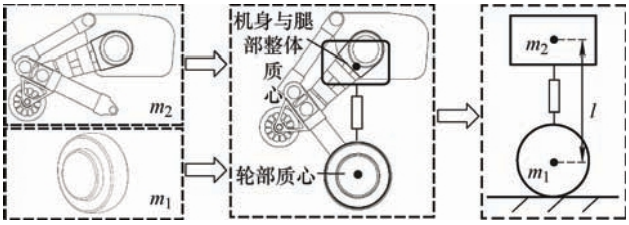


Fig. 5 Nonlinear spring dynamics model for a two-mass block

Since the above virtual model is only used in the jumping phase and landing phase during the trajectory planning process in this paper, at this time, the lower mass block is always in contact with the ground, and the height and velocity are both 0, so only the dynamics model of the upper mass block needs to be considered as shown in Equation (10)

$$m_{22}\ddot{z}(t) = a[l_0 - z(t)]^3 + b[l_0 - z(t)] - m_2 g \quad (10)$$

where l_0 is the original length of the nonlinear spring, and a, b are the nonlinear spring strength coefficients.

1.2.3 Single leg static model

The purpose of modeling single-leg statics is to incorporate a two-mass block nonlinearly is mapped to the hip space so that the leg model

The support force of the spring in the leg

The ability to track the virtual model to plan trajectories to accomplish jumping maneuvers. Due to the high acceleration of the wheels during high jumps in place and jumping maneuvers while traveling, the wheel acceleration is very high.

coordinates of each connecting rod in the

coordinate system of Fig. 4

$$\begin{cases} x_{m1} = L_{m1}\cos(\theta_1 - \beta_{m1}) \\ z = L\sin(\theta - \beta) \end{cases}$$

$$\begin{cases} x_{m2} = L_{11} + L_{m2}\cos(\theta_2 - \theta_1 - \beta_{m2}) \\ z = L\sin(\theta - \theta - \beta) \end{cases}$$

$$\begin{cases} x_{m3} = L_1 + L_{m3}\cos(\theta_3 - \theta_1 - \beta_{m3}) \\ z_{m3} = L_{m3}\sin(\theta_{03} - \theta_{01} - \beta_{m3}) \\ x_{m4} = L_{11} + L_2\cos(\theta_2 - \theta_1) - L_{m4}\cos\theta_5 \\ z = L\sin(\theta - \theta) - L\sin\theta \end{cases} \quad (7)$$

where $\theta_5 = \theta_2 + \theta_4 - \theta_1 + \beta_{m4}$.

In turn, the coordinates of the upper mass block a

equation (8)

$$\begin{cases} x_{ms} = \dfrac{2mL1\,x_{m1} + 2mL2\,x_{m2} + 2mL3\,x_{m3} + mL4\,x_{m4}}{2mL1 + 2mL2 + 2mL3 + mL4} \\ z_{ms} = \dfrac{2mL1\,z_{m1} + 2mL2\,z_{m2} + 2mL3\,z_{m3} + mL4\,z_{m4}}{2mL1 + 2mL2 + 2mL3 + mL4} \end{cases} \quad (8)$$

After that, the derivation of Eq. (8) and substitution of Eq. (6) to obtain the height difference between the center of mass of the upper and lower mass blocks and its rate of change are shown in Eq. (9)

is small and the mass of the leg bars is light, so the effects of bar moment of inertia as well as wheel acceleration are ignored to simplify the calculations. By means of the single-leg Static modeling to analyze the force between the rods in the self-equilibrium state of the wheel foot

force cases, to establish the support between the joint outflow and the mass block at different heights

The force relationship is shown in Fig. 6.

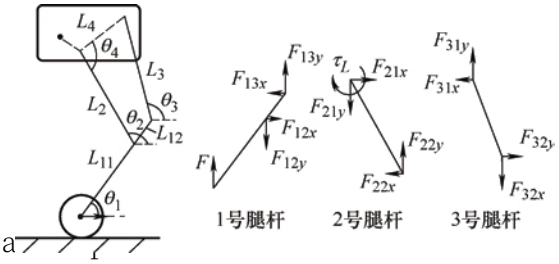


Fig. 6 Single leg static model

Neglecting the change in fuselage pitch angle in balance control due to fuselage with the overall center of mass of the leg always directly above the center of the wheel, such that equation (8)

In $x_{ms} = 0$ it is possible to find the angle θ_1 between the leg bar and the ground for leg 1. In order to square

For ease of calculation, fitting the relationship yields equation (11)

$$\theta_4(\theta_1) = 0.2510\theta_1^3 - 0.9010\theta_1^2 + 1.9330\theta_1 - 0.9329 \quad (11)$$

Firstly, the force analysis of leg bar No. 1 is shown in equation (12)

$$\begin{cases} F_{12x} = F_{13x} \\ F + F_{13y} = F_{12y} \\ F_{13y} \tan[\theta_{14}(\theta) + \theta_{24}(\theta)] = F_{13x} \\ F_{13x} L_{12} \tan\theta_1(\theta_4) + F_{13y} L_{12} = F \cdot L_{11} \end{cases} \tag{12}$$

From Eq. (12), we can obtain

$$F_{13x} = -F \cdot L_{11} \cot[\theta_{01}(\theta_4) + \theta_{03}(\theta_4)] / X$$

$$F_{13y} = F \cdot L_{11} / X$$

$$F_{12x} = F_{22x} = -F \cdot L_{11} \cot[\theta_{01}(\theta_4) + \theta_{03}(\theta_4)] / X$$

$$F_{12y} = F_{22y} = FX \cdot$$

$$\{L_1 - L_{12} \cot[\theta_{01}(\theta_4) + \theta_{03}(\theta_4)] \tan\theta_{01}(\theta_4)\} \tag{13}$$

where $X = L_{12}\{1 - \cot[\theta_1(\theta_4) + \theta_3(\theta_4)] \tan\theta_1(\theta_4)\}$. After that, the force analysis of leg bar No. 2 is shown in Eq. (14)

$$\tau_L = F_{22y} L_2 \sin\theta_6 - F_{22x} L_2 \cos\theta_6 \tag{14}$$

where $\theta_6 = \theta_1 + \theta_2 - \pi/2$.

Substituting Eq. (13) into Eq. (14) the relationship between the support force between the mass blocks and the hip joint moment can be found as shown in Eq. (15)

$$\tau_L = F_{L2}[L_{12}\cot(\theta_1 + \theta_3)\tan\theta_1 + L_{11}\cot(\theta_1 + \theta_3)\tan(\theta_{01} + \theta_{02}) - L_1]/[L_{11}\cot(\theta_{01} + \theta_{03})\tan(\theta_{01} + \theta_{02})] \tag{15}$$

1.2.4 Modeling of airborne dynamics

The relationship between the output moment of the wheel part and the pitch angle acceleration of the fuselage during the robot's vacating is established by modeling the vacating dynamics.

where $a_{11} = \frac{g}{l(\theta_4)} + l_2(\theta_4)$.

2 Jumping Algorithm Based on Virtual Models

In this section, the trajectory planning for each jump phase delineated in Fig. 1 is first performed in the virtual model space to obtain the spring strength coefficients a_1, b_1, a_2, b_2 *in the jumping and landing phases* as well as the spring strength coefficients a, b, a, b in the vacating phase and the landing preparation phase.

The acceleration curves $\ddot{z}_1(t)$, $\ddot{z}_2(t)$ of the upper and lower mass blocks, and then use the aforementioned Parameter solving upper and lower center of mass trajectories as a function of time $z_1(t), z_2(t)$, introducing the mapping between the virtual model and the robot model relationship and designed the trajectory tracking control algorithm.

2.1 Virtual model jump trajectory planning

2.1.1 Regiment body phase trajectory planning

The jump trajectory planning method designed in this paper starts from the doughnut phase of the

The corpuscular phase trajectory ensures that the overall center of mass of the robot reaches its highest point when the upper

Lower mass blocks all reach preset heights

$$\begin{cases} \ddot{z}_1(t) = d_1 t^3 + d_2 t^2 + d_3 t \\ \ddot{z}_2(t) = \dfrac{-m_1 \ddot{z}_1(t) - m_1 g - m_2 g}{m_2} \end{cases} \tag{17}$$

If the nonlinear spring model is based on a two-mass block, it is difficult to find the rationale for the

The desired trajectory and the maximum height that can be reached by the wheel part are limited. Therefore, in this paper, we use the cubic interpolation method for the acceleration of the mass block under the body phase of the regiment.

Planning is carried out as shown in equation (17). Assuming that the overall center of mass of the robot is high

The degree change follows a parabolic trajectory, then the time experienced by the

body phase of the regiment is The wheel algorithm is designed to control the pitch angle of the body of the vacating phase, which in turn realizes the attitude stabilization and in-situ jumping function of the robot, as shown in Fig. 7 Shown.

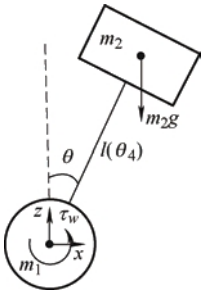


Fig. 7 Modeling of airborne dynamics

In the vacating dynamics model, the robot is also simplified into two mass blocks with variable distance, the influence of leg movements on the body pitch angle is ignored, and the center of rotation of the wheel is used as the coordinate origin to establish the vacating dynamics The academic model is shown in equation (16)

$$\begin{bmatrix} \ddot{\theta} \\ \ddot{l} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{l} \end{bmatrix} + \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \tau_l \\ \tau_r \end{bmatrix} \tag{16}$$

$t_{12} = t_2 - t_1 =$ The constraints are as follows

$$\begin{cases} z_2(t_2) - z_2(t_1) = 2l_s \\ \dot{z}_{12}(t) = 0 \end{cases} \tag{18}$$

$$\begin{cases} z_2(t) = l(t) + z_s \\ \dot{z}_2(t_2) = 0 \end{cases}$$

According to Eq. (18), the three design parameters d_1, d_2, d_3 in Eq. (17) as well as the upward velocity $\dot{z}_2(t_1)$ of the upper mass block at the initial moment of the mass phase can be obtained.

2.1.2 Jump phase trajectory planning

According to the results of the trajectory planning of the doughnut phase, the center of mass trajectory of the starting phase is planned based on the nonlinear spring model. In order to facilitate the robot to realize continuous jumping, given the spring length $l_s(t_0)$ at the initial moment of the jumping phase and the spring length $l_s(t_1)$ at the end moment of the jumping phase, the original length of the spring of the jumping phase l_1, the spring strength coefficients a_1 and b_1 are taken as the design parameters, which are shown in equation (19)

$$m g[z_2(t) - z_{12}(t)] + \frac{1}{2} m \dot{z}_{21}^2(t) = \frac{1}{4} a_1 \{[l - z_{12}(t)]^4 - [z_2(t) - l]^4\} + \frac{1}{2} b_1 \{[l - z_{12}(t)]^2 - [z_{02}(t) - l]^2\}$$

$$m g = a_1 [z_2(t) - l]^3 + b_1 [z_2(t) - l]$$

$$F = a_{max1}[l - z_{12}(t)]^3 + b_{01}[l - z_{120}(t)] \quad (19)$$

The first equation ensures that the upper mass block is available at the moment of the end of the starting phase to obtain a set rate of ascent $z_{21}(t)$; the second equation ensures that the starting jump

At the end of the phase the spring tension is balanced by the gravity of the lower mass of the block; the third and other style constrains the maximum spring support force and ensures that the hip joint moment does not exceed

over the load capacity of the motor. If a linear spring is used to plan the trajectory there is no support force while guaranteeing $l(t)$ if $l_{s1}(t)$

The method constrains the maximum l_{s1}

As a design parameter, the maximum height of the mass on the constraint needs to be dropped

$z_2(t_2)$, demonstrating the flexibility of nonlinear springs in trajectory planning.

2.1.3 Landing preparation phase trajectory planning

In order to minimize the impact on the wheels of the robot when landing, the landing

The planning goal for the preparation phase is to make the final velocity of the lower mass block $z_{11}(t_3)$

is 0, while the height of the lower mass during the fall cannot be less than

0. The trajectory is also planned using the cubic interpolation method, see equation (20)

$$\begin{cases} z_1(t) = d_4 t^3 + d_5 t^2 + d_6 t \\ z_{12}(t) = \dfrac{-m_{11} z_1(t) - m_1 g - m_2 g}{m_2} \end{cases}$$

$$\begin{cases} z_1(t_3) = 0 \\ z_{13}(t) = 0 \end{cases}$$

$$\text{s.t.} \quad \begin{cases} z_{12}(t_3) = l_s(t_3) \\ z_2(t) = -z_{321}(t) \\ z_{21}(t) \geq 0 \end{cases} \quad (20)$$

2.1.4 Landing phase trajectory planning

The goal of the landing phase trajectory

$$\begin{cases} t_{12} = \sqrt{\dfrac{2[z_s - z_{13}(t_1)]}{g}} \\ m\sqrt{2g[z_s - z_{13}(t_1)]} \end{cases}$$

$$\begin{cases} z_{31}(t) = \dfrac{2}{m_1 + m_2} \\ x_{31}(t) = \dfrac{x_{2t12}}{x_3(t_1)} \end{cases} \quad (22)$$

$$\beta_b = \arctan\left[\frac{z_{31}(t)}{}\right]$$

Pitch needs to be taken into account in the planning of the trajectory of the jump phase and the landing phase.

The effect of angular bias is shown in equation (23)

$$m_2 g[z_{12}(t) - z_{120}(t)]\cos\theta_b + \frac{1}{2} m_2 z_{221}^2(t) = \frac{1}{4} a_{max1}\{[l - z_{120}(t)]^4 - [z_{12}(t) - l]^4\} + \frac{1}{2} b_{01}\{[l - z_{120}(t)]^2 - [z_{12}(t) - l]\}^2$$

$$m_2 g = \{a_{max1}[z_{12}(t) - l]^3 + b_{01}[z_{12}(t) - l]\}\cos\theta_b$$

$$F = a_{max1}[l - z_{112}(t)]^3 + b_{01}[l - z_{120}(t)] \quad (23)$$

In addition, the pitch attitude of the fuselage during the airborne process is controlled by the wheel section during the in-situ long jump, and the target pitch angle for each of the four phases needs to be set separately, and at the same time, in order to minimize the error of the stalling time, the landing

The fuselage pitching attitude is symmetrical to that of the jump, see equation (24).

$$\theta_s(t) = \begin{cases} \theta_{t1}(\theta_0) + \theta_b, t \in [t_0, t_1] \\ \theta_{t2}(\theta_4), t \in [t_{12}, t_2] \\ \theta_{t3}(\theta_4) - \theta_b, t \in [t_{23}, t_3] \end{cases} \quad (24)$$

planning is to keep the robot's attitude at the end of the phase consistent with the initial moment

of the jump phase so that the robot can realize continuous jumps. The velocity $\dot{z}_1(t_4)$ of the lower mass block at the end of the landing preparation phase is 0. At the same time, when the upper mass block reaches the initial jump height $z_2(t_4)$, its velocity $\dot{z}_2(t_4)$ also decays to 0. Therefore, the trajectory planning of the landing phase is shown in Equation (21)

$$\begin{cases} m_1 g = a_1[l_1(t) - l_{s1}]^3 + b_1[l_1(t) - l_{s1}] \\ m_1 g[z_1(t) - z_2(t)] + \frac{1}{2} m_1 \dot{z}_1^2(t) = \frac{1}{4} a_1\{[l_1 - l_1(t)]^4 - [l_1 - l_{s1}]^4\} + \frac{1}{2} b_1\{[l_1 - l_1(t)]^2 - [l_1 - l_{s1}]^2\}^2 \\ F_{\max 2} = a_2[l_2 - l_2(t)]^3 + b_2[l_2 - l_2(t)] \end{cases} \tag{21}$$

2.1.5 Planning the trajectory of the long jump in situ

Assuming that there is no initial moving speed of the wheel at the time of jumping and there is no leg retracting action after vacating, the initial z-direction speed and stagnation time can be obtained according to the given jumping height z_s, and the initial x-direction speed can be obtained by combining with the given jumping distance to find the pitch angle at the time of jumping, as shown in Eq. (22).

$$\begin{cases} \dot{z}_1(t_4) = 0 \\ \dot{z}_2(t_4) = 0 \\ \theta_f(\theta_4), t \in (t_3, t_4) \end{cases} \tag{22}$$

2.2 Leg Trajectory Tracking Algorithm

2.2.1 Virtual model upper and lower center of mass trajectory solution

In the regimentation and landing preparation phases, the result of trajectory planning is a curve of acceleration versus time for the upper and lower mass blocks, and a quadratic integration of this curve yields the planned trajectory. However, in the jumping and landing phases, the result of the planning is the two coefficients of strength of the nonlinear spring.

According to the established nonlinear spring dynamics model, the planning trajectory is obtained

Second order nonlinear differential equations need to be solved.

Firstly, the reduced order treatment of Eq. (7) yields

$$\frac{1}{2} m_2 p^2 = C - \frac{1}{4} a_2[l_{02} - z_2(t)]^4 - \frac{1}{2} b_2[l_{02} - z_2(t)]^2 - m_2 g z_2(t) \tag{25}$$

where $p = \mathrm{d}z_2(t)/\mathrm{d}t$.

Substitute the initial length of the spring to get the constant term C, and then organize the equation (25) to get

$$\dot{z}_2(t) = \sqrt{\frac{4C - a_2(l_{02} - z_2(t))^4 - 2b_2(l_{02} - z_2(t))^2}{2m_2} - 2gz_2(t)} \tag{26}$$

The planning trajectory can be obtained by iterating Eq. (26).

2.2.2 Virtual model space and joint space mapping relationship

In the previous paper, the mapping relationship between the hip joint angle, angular velocity and the height difference between the center of mass of the upper and lower mass blocks and their changing velocities was established based on the kinematic model, which is detailed in Eq. (9); and the hip joint force was established based on the single-leg kinetic model

The mapping relationship between the moments and the support force between the upper and lower mass blocks is detailed in Eq. (15).

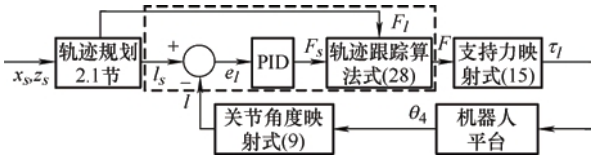
$$l(\theta_4) = 0.1098\theta_4^3 - 0.5559\theta_4^2 + 1.128\theta_4 - 0.6153$$

$$\theta_t(\theta_4) = -3.097\theta_4^5 + 22.11\theta_4^4 - 63.671\theta_4^3 + 92.467\theta_4^2 - 67.828\theta_4 + 20.1129$$

$$F(\theta_4) = \tau(-31.959\theta_4^5 + 199.407\theta_4^4 - 489.44\theta_4^3 + 585.907\theta_4^2 - 330.942\theta_4 + 71.8476)$$

$$\tau(\theta_4) = F\cdot(-0.4341\theta_4^5 + 3.622\theta_4^4 - 11.875\theta_4^3 + 19.296\theta_4^2 - 15.722\theta_4 + 5.3206) \quad (27)$$

However, since the robot legs used in this paper have parallel connections



structure, it is difficult to map the speed of change of the height difference of the center of mass to the joint space, so the hip joint angle and angular velocity are mapped to the virtual model space, and the trajectory tracking algorithm is established in the virtual model space to obtain the target support force between the mass blocks and then mapped to the hip joint space. Due to the complexity of the analyzed results and because the leg is a single-degree-of-freedom structure, the fitting results have a high consistency with the analyzed results. Therefore, in order to reduce the computational cost, the fitting results are utilized instead of the analytical results in the simulation process, as shown in Eq. (27).

$$F_m = \begin{cases} m_2 g, & t \in [t_0, t_1] \cup [t_3, t_4] \\ -m g, & t \in (t_1, t_3) \end{cases} \quad \text{s.t.} \quad (28)$$

3 jump simulation

The correctness of the above trajectory planning algorithm is verified by Simulink-Adams co-simulation. Firstly, the 3D model of the wheel-legged robot is imported into Adams for pre-processing, and the obtained dynamics model is imported into Simulink to establish the robot control system.

3.1 Jump simulation without closing the legs

Tracks based on linear and nonlinear spring models, respectively

The trajectory planning results were simulated for jumps without leg retraction, comparing the two models up to the

The demand for force on the hip joint when going to the same height over the obstacle.

Setting goals

2.2.3 Trajectory tracking control algorithm

In this paper, three trajectory tracking control algorithms are designed, and the specific implementation method is shown in Fig. 8.

Fig. 8 Trajectory tracking control algorithm

The first one is trajectory mapping control algorithm, which maps the hip joint angle to the virtual model space, and the support force between the mass blocks is calculated by the trajectory tracking plus feed-forward moment compensation algorithm and mapped to the hip joint space to realize the jump; the second one is force mapping control algorithm, which is the support force between the mass blocks in the planning result is directly mapped to the hip joint space to realize the jump; and the third one is the force-position mixing control

algorithm, which sums the support force obtained by the trajectory tracking algorithm in the virtual model and the planning support force to realize the jump, as shown in Equation (28). The third is the force-position hybrid control algorithm, which maps the support force obtained by the trajectory tracking algorithm in the model and the planned support force to the hip joint space to realize the jump, as shown in Eq. (28).

$$\begin{cases} \text{Position mapping method: } F = F_s + F_m \\ \quad a[l(t) - l]^3 + b[l(t) - l], \\ \text{force mapping method: } f = f = \begin{cases} 0, & t \in [t, t] \cup [t, t] \\ -mg - m\ddot{z}(t), & t \in (t, t) \end{cases} \\ \text{force-position mixing method: } F = F + F \end{cases} \quad (28)$$

The crossing heights were 0.02 m, 0.04 m, and 0.06 m, respectively, according to pp.

The method described in section 2.1.2 is used to plan the jumping phase, the target distance between the upper and lower mass blocks is kept constant at the end of the jumping phase, and the force-position mixing method is used for the jumping control as shown in Fig. 9, and the results of the jumping are shown in Table

4 Shown.

Fig. 9 Simulation effect of jumping with wheeled legs without retracting legs

Table 4 Comparison of jumping effects of two spring models

Spring-loaded barrier height		Coefficient of restitution (in Hooke's law)	coefficient of restitution (in Hooke's law)	maximum support	Actual height
Type	degree/m	/(N · m)$^{-1}$	/(N · m)$^{-1}$	Force/N	Degree/m
sprung	0.02	-	3 028.2	265.68	0.020 1
Linear	0.04	-	3 714.2	334.28	0.040 1
	0.06	-	4 400.2	402.88	0.060 2
	0.02	-307 740	4 080.6	140	0.020 2
Non-linear sexual orientation (e.g. gay)	0.04	-384 000	5 140	170	0.040 2
	0.06	-461 140	6 202	200	0.060 5
Reed	0.16	-742 000	11 100	400	0.161 1

As shown in Table 4, the control error of the wheel crossing height is within 1%. Under the condition of the same obstacle-crossing height, the maximum support force required by the nonlinear spring is much smaller than that of the linear spring, and the maximum obstacle-crossing height of the robot is set to be 0.06 m. The linear spring model requires the legs of the robot to be 0.06 m high.

The nonlinear spring model provides a support force of 402.88 N, while the nonlinear spring model only

A support force of 200 N is required to further set the nonlinear spring

The robot's obstacle-crossing height at a maximum support force of 400 N for the spring model

The degree of the linear spring model reaches 0.16 m, which is much larger than that of the linear spring model of 0.06 m. The degree of the

linear spring model roaches 0.16 m

l s l

crossing height, proving that the dual mass block nonlinear spring model can be Significantly reduces the maximum force out of the hip joint and improves the motor torque of the li Usage rate.

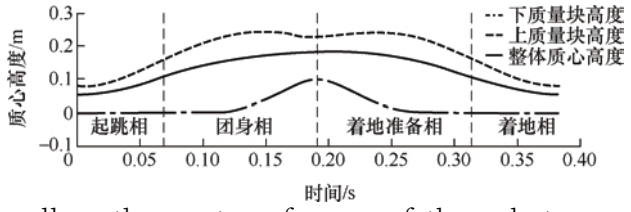
3.2 Complete Planning Trajectory Jump Simulation

In this section, the complete jump trajectory of the nonlinear spring model is planned using the barrier height of 0.1 m as an example, and the trajectory planning constraints are shown in Table 5.

Table 5 Trajectory planning constraints

	$z_1(t)$ /m	$z_2(t)$ /m	$z_3(t)$ /m	$z_1(t)$ /(m · s^{-1})	$z_2(t)$ /(m · s^{-1})	$z_3(t)$ /(m · s^{-1})
$t\,0_0$		0.080	0.054	0	0	0
$t\,0_1$		0.160	0.107	0	1.784	1.197
t_2	0.100	0.220	0.181	0	0	0
$t\,0_3$		0.160	0.107	0	-1.784	-1.197
$t\,0_4$		0.080	0.054	0	0	0

The jump trajectories are calculated based on the planning parameters in Table 5, and the trajectory planning results are finally fitted with time as the horizontal coordinate to obtain the height trajectories of the center of mass of the upper and lower mass blocks as well as the center of mass of the robot as a whole in Fig. 10, and the support force between the mass blocks is shown in Fig. 11.



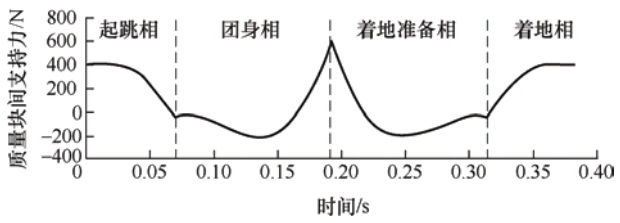


Fig. 10 Planned trajectory curves of the center of mass

Fig. 11 Planning curve of support force between mass blocks

The complete trajectory is planned for the trajectory of the wheel part of the vacating phase, which further improves the robot's crossing height, and the overall effect is shown in Fig. 12.

Fig. 12 Simulation effect of jumping with complete planning trajectory

The three trajectory tracking methods described in the previous section are used to realize the jumping action of the robot at different control frequencies, respectively, and the results are shown in Table 6.

Table 6 Comparison of deviation of trajectory tracking algorithms under different control frequencies

Control frequency/Hz	$z_3(t_2)$ /m	$z_2(t_2)$ /m	$z_1(t_2)$ /m	$z_3(t_3)$ /m	$z_1(t_3)$ /m
Position Mapping	1000.007	10.006	00.009	50.031	90.015 9
	2000.003	30.002	70.004	80.025	10.009 1
	3000.002	20.001	40.004	10.022	50.006 9
	5000.001	10.000	90.001	60.021	00.004 7
Power Reflections	1000.040	90.039	40.044	00.076	60.099 1
		2000.027	60.028	40.026	10.044
	80.036 9				
Shot	3000.022	50.026	40.014	60.037	40.021 4
	5000.019	20.026	50.004	30.024	30.037 4

	1000.018	00.017	80.018	50.029	70.018 1
Force Levels	2000.009	40.010	40.007	40.	012
	90.001 7				
Mixed	3000.009	80.011	10.007	40.014	10.003 5
	5000.005	60.007	10.002	60.014	40.006 7

The simulation results verify the correctness of the trajectory planning method in this paper. According to Table 6, when comparing the control accuracy of the three trajectory tracking methods under different control frequencies, it can be seen that the control accuracy of force mapping method is the lowest among the three control methods, the control accuracy of position mapping method is the highest for the height of the wheel part, and the integrated control accuracy of force-position hybrid control is the highest for the height of each center of mass. When the control frequency is greater than 100 Hz, the overall error of position mapping and force-position hybrid control can be controlled within 2 cm, and the error of wheel height can be controlled within 8 mm; when the control frequency reaches 500 Hz, the error of wheel height control is within 3 mm. All three methods can jump over the set obstacle height, if only need to jump over the specified height, the position mapping method is the most effective; if need to limit the maximum height of the fuselage during the jumping process, the force-position hybrid control algorithm is more effective, and the control frequency should be more than 200 Hz.

3.3 Jumping Simulation on the Move

The wheel-legged robot jumps in the process of traveling with the same trajectory planning as in Section 3.2, according to the given height of the obstacle to calculate the leg planning trajectory and the robot stagnation time, combined with the given jumping distance can be found when the required speed of the jump. In the simulation, the robot first reaches the preset speed, and when the robot enters the state of uniform motion, the trajectory tracking algorithm is used to control the legs to realize the robot jumping while traveling, and the overall effect is shown in Fig. 13.

Fig. 13 Simulation effect of jumping while traveling

The jump height and distance of the robot were set to be 0.1 m According to the The jump height was calculated to be 0.244 4 s for the jump duration, which was then combined with the jump The jump distance can be obtained that the initial moving speed should be 0.409 2 m/s. Simulation open

After starting, the robot first uses the leg position control to bring the spring length up to the level of 0.08 m, set the target moving speed, when the robot finished accelerating into the state of uniform motion, the position mapping method was used to realize the jumping action, and the jumping height and distance of the robot were recorded, as shown in Table 7.

Table 7 Simulation results of jumping while traveling

	$_{xs}$/m	$\dot{x}(t)$/(m · s)$^{-1}$ $_{31}$	$_{z1}(_{t2})$/m	$_{z1}(_{t3})$/m
Setting value	20.100 0	0.100 0	00.409	
actual value	0.134 7	0.413 2	0.102 2	0.005 5
deviation	0.034 7	0.004 0	0.002 2	0.005 5
Deviation rate (%)	34.700	0.978	2.200	-

From Table 7, it can be seen that the prediction accuracy of jumping in traveling is within 3.5 cm for jumping distance, and the prediction accuracy for crossing height is between 2.5 mm or less. Comparing with the results in Table 6, the results of jumping height in traveling and in situ jumping height are basically the same, which indicates that when the robot jumps at a uniform speed, the wheel action and leg jumping action are decoupled, and the traveling speed does not affect the robot's jumping height.

3.4 Long Jump Simulation

The wheel-legged in-situ jumping algorithm proposed in this paper makes the robot jump in the state of forward tilt of the center of mass, and the body pitch attitude is adjusted in accordance with the established target value through the wheel control in the process of vacating, and the overall jumping effect is shown in Fig. 14.

Fig. 14 Simulation effect of long jump in place

From Fig. 14, it can be seen that the wheel part can play a stabilizing role in the process of robot vacating, and the single-loop PID control algorithm can realize the accurate control of the pitch angle of the fuselage in the vacating phase. However, from Table 8, it can be seen that the robot will accelerate in one direction after offsetting the target pitch angle of the fuselage, and when the jumping action is turned on, the robot has a

Provides some initial movement speed, which severely affects jumping distance

The accuracy of the jump if the effect of the initial movement speed is removed

The deviation of the distance was 3.9 cm, and also compared to jumping in place, the more

The control error for barrier height increased to 1.11 cm. However, compared to the

Jumping while traveling, the in-place jump control algorithm designed in this paper implements the

Taking the jump height of 0.1 m and the jump distance of 0.05 m as an example, the robot does not retract its legs during the vacating process, and uses force-position hybrid control to realize the jumping action, and the trajectory planning and simulation results are shown in Table 8.

Table 8 Simulation results of long jump in place

parameters	set point	actual value	deviation	deviation rate (%)
$z(t)$/m	0.100 0	0.111 1	0.011 1	11.10%
$z_2(t_2)$/m	0.280 0	0.297 4	0.017 4	6.21%
$z_3(t_2)$/m	0.220 8	0.236 2	0.015 4	6.94%
$z_1(t_3)$/m	0.000 0	0.001 1	0.001 1	-
$z_2(t_3)$/m	0.180 0	0.183 4	0.003 4	1.89%
$z_3(t_3)$/m	0.120 8	0.123 5	0.002 6	2.18%
t_{13}/s	0.299 2	0.299 0	-0.000 2	-0.07%
x_s/m	0.050 0	0.179 4	0.129 4	258.80%
$\dot{x}(t)$/(m · s^{-1})	0.000 0	0.299 3	0.299 3	-

12

31

has a greater jumping distance, which is important for improving the robot's jumping performance.

Significance.

4 Conclusion

(1) A jump trajectory planning method based on a nonlinear spring model with two mass blocks is proposed, in which the support force is mapped to the hip joint space to realize the jumping action after trajectory tracking control in the virtual model space. Compared with the linear spring-based trajectory planning method, this method is not only more flexible in trajectory planning, but also can realize higher jumping height under the same condition of hip joint force.

(2) The attitude control algorithm based on the vacuolar dynamics model is designed not only to ensure that the body pitch angle is always controllable during the jumping process and to improve the jumping stability of the single-degree-of-freedom wheel-legged robot, but also to further utilize the role of the wheel part in controlling the attitude to design the algorithm of the in-situ jumping, so that the robot can achieve the same jumping distance within a shorter jumping distance and jumping time.

(3) A virtual prototype model was established, and simulation tests of non-retractable jump, in-situ jump, traveling jump and in-situ long jump were carried out to verify the feasibility of the trajectory planning and tracking algorithms. The simulation results show that when the control frequency is greater than 200 Hz, the position mapping method and the force-position hybrid method can ensure that the wheel height error is within 3 mm, but the control accuracy of the in-situ long jump needs to be improved.

refer to Examining Literature Literature

[1] WANG Buyun, PENG Jian, LIANG Yi, et al. Design and characterization of suspension mechanism for all-terrain mobile robot[J]. Journal of Mechanical Engineering, 2022, 58(9): 71-86.

WANG Buyun , PENG Wen , LIANG Yi , et al. Characteristics analysis and optimization design of Suspension mechanism of all-terrain mobile robot[J]. Journal of Mechanical Engineering, 2022, 58(9):71-86.

[2] QIANG R, WU J, YAO Y A. Design and analysis of a multi-legged robot with pitch adjustive units[J]. Chinese Journal of Mechanical Engineering, 2021, 34(3): 64-81.

[3] ZHENG Y, XU K, TIAN Y, et al. Bionic design and analysis of a novel quadruped robot with a multistage buffer system[J]. Chinese Journal of Mechanical Engineering, 2022, 35(2):32-52.

[4] Boston Dynamics. about handle[OL]. (2019-11-22) [2019-11-25]. Available from: //www.bostonDynamics.com/handle.

[5] KLEMM V, MORRA A, SALZMANN C, et al. Ascento: A two-wheeled jumping robot[C]// 2019 International Conference on Robotics and Automation (ICRA). ieee, 2019:515-7521.

[6] KLEMM V, MORRA A, GULICH L, et al. LQR-Assisted whole-body control of a wheeled bipedal robot with kinematic loops[J]. IEEE Robotics and Automation Letters, 2020, 5(2):3745-3752.

[7] CUI L, WANG S, ZHANG J, et al. Learning-Based Balance control of wheel-legged robots[J]. IEEE Robotics and Automation Letters, 2021, 6(4):7667-7674.

[8] Wang Zhennan. Design and vertical jump control study of hydraulic single-leg robot [D]. Harbin: Harbin Institute of Technology, 2018.
WANG Zhennan. design of hydraulic single leg robot and research on vertical jump control[D]. Harbin: Harbin Institute of Technology, 2018.

[9] Gao Desheng. Pneumatic double-jointed bouncing leg jumping mechanism and experimental research [D]. Harbin: Harbin Institute of Technology, 2021.
GAO Desheng. Jumping mechanism and experimental study of pneumatic double joint spring leg[D]. Harbin: Harbin Institute of Technology, 2021.

[10] HUTTER M, REMY C D, HOEPFLINGER M A, et al. High compliant series elastic actuation for the robotic leg ScarlETH[M]. World Scientific Publishing, 2011.

[11] C.W. Chen. Research on control system of hydraulically driven single-legged jumping robot [D]. Hang State: Zhejiang University, 2016.
CHEN Zhiwei. Research on control system of hydraulically actuated single-legged hopping robot[D]. Hangzhou: Zhejiang University, 2016.

[12] TERRY P, PIOVAN G, BYL K. Towards precise control of hoppers: using high order partial feedback linearization to control the hopping robot FRANK[C]// 2016 IEEE 55th Conference on Decision and Control (CDC). ieee, 2016: 6669-6675.

[13] semini c, tsagarakis n g, guglielmino e. et al. Design of HyQ - a hydraulically and electrically

actuated quadruped robot[J]. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2011, 225(6): 831-849.

[14] MATHIS F B, MUKHERJEE R. Apex height control of a two-mass robot hopping on a rigid foundation[J]. Mechanism & Machine Theory, 2016, 105: 44-57.

[15] Shan Kaizheng, Yu Haitao, Han Liangliang, et al. Approximate direct-drive bipedal robot jumping Nonlinear optimization and experimental validation of jump motion[J]. Journal of Mechanical Engineering. 2021, 57(13):153-162.
SHAN Kaizheng, YU Haitao, HAN Liangliang, et al. Nonlinear optimization and experimental validation of a Quasi-direct-drive bipedal robot's jumping motion[J]. Journal of Mechanical Engineering , 2021 , 57(13): 153-162.

[16] YU Jianjun, ZHANG Yuan, ZUO Guoyu, et al. Gait imitation of humanoid robot based on ZMP criterion[J]. Journal of Beijing Institute of Technology, 2018, 44(9): 1187-1192.
YU Jianjun, ZHANG Yuan, ZUO Guoyu, et al. Humanoid robot gait imitation based on ZMP criterion[J]. Journal of Beijing University of Technology , 2018 , 44(9) : 1187-1192.

[17] Fan Yixiao. Research on jump control of wheel-legged robot [D]. Harbin: Harbin Institute of Technology, 2020.
FAN Yixiao. Research on the jumping control of the wheel-legged robot[D]. Harbin ： Harbin Institute of Technology, 2020.

[18] Li X, FAN Y, YU H, et al. Stable jump control for the wheel-Legged robot based on TMS-DIP model[J]. Industrial Robot, 2022, 49(2):212-225.

[19] Yu 溳陽. Research on jumping control of hydraulic biped wheel-legged robot [D]. Harbin Bin: Harbin Institute of Technology, 2021.
YU Haoyang. Research on the jump control of hydraulic wheel-leg biped robot[D]. Harbin: Harbin Institute of Technology, 2021.

[20] CHEN H, WANG B, HONG Z, et al. Underactuated motion planning and control for jumping with wheeled-bipedal robots[J]. IEEE Robotics and Automation Letters, 2020, 6(2):747-754.

[21] Luo Yao, Wang Hui. Research on nonlinear spring damping vibration damping device[J]. Physical and Engineering, 2011, 21(5):4.
LUO Yao, WANG Hui. Study of the nonlinear damping spring shock absorbers[J]. Physics and Engineering , 2011, 21(5):4.

[22] Cai Chunshan, Wang Zuoxun. Balance control of two-wheeled robot based on LQR[J]. Journal of Qilu University of Technology, 2018, 32(1):55-60.

CAI Chunshan , WANG Zuoxun. Balance control of two-wheeled robot based on LQR[J]. Journal of Qilu University of Technology, 2018, 32(1):55-60.

[23] Wen Shuangyun. Research and optimization of control algorithm for two-wheeled self-balancing trolley [D]. Changchun: Jilin University, 2017.

WEN Shuangyun. the research and optimization on control algorithm of two-wheeled self-balance vehicle[D]. ofChangchun: Jilin University, 2017.

Author：Jingsong Gao, male, born in 1995, is a doctoral student. His main research interests are wheel-legged self-balancing robots.

E-mail: 19B908036@stu.hit.edu.cn

Hongchul Kim (Corresponding Author), male, born in 1976, Ph. His main research interests include solo wheel robot, variable stiffness flexible joint design and its control theory, and multi-redundancy robot manipulator arm.

E-mail: hongzhejin@hit.edu.cn