

Boxing WANG, Chunlin ZHOU, Ziheng DUAN, Qichao ZHU, Jun WU, Rong XIONG

Untethered quadrupedal hopping and bounding on a trampoline

© The Author(s) 2019. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract For quadruped robots with springy legs, a successful jump usually requires both suitable elastic parts and well-designed control algorithms. However, these two problems are mutually restricted and hard to solve at the same time. In this study, we attempt to solve the problem of controller design with the help of a robot without any elastic mounted parts, in which the untethered robot is made to jump on a trampoline. The differences between jumping on hard surfaces with springy legs and jumping on springy surfaces with rigid legs are briefly discussed. An intuitive control law is proposed to balance foot contact forces; in this manner, excessive pitch oscillation during hopping or bounding can be avoided. Hopping height is controlled by tuning the time delay of the leg stretch. Together with other motion generators based on kinematic law, the robot can perform translational and rotational movements while hopping or bounding on the trampoline. Experiments are conducted to validate the effectiveness of the proposed control framework.

Keywords hopping and bounding gait, compliant mechanism, compliant contact, balance control strategy, legged locomotion control, quadruped robot

1 Introduction

Hopping or bounding, which is one of the fundamental abilities of quadruped robots, can help these robots

overcome obstacles and increase movement speed. WildCat, Spot, and Spotmini, which were developed by Boston Dynamics, demonstrated how robots can attain remarkable agility from dynamic bounding or running. MIT Cheetah 1 and 2 took advantage of bounding gait to achieve high-speed movement and obstacle clearance jumping [1–3]. StarLETH [4] and ANYmal [5] with pronking gaits are other exemplary results of the advancements made in quadrupedal robotic systems.

Various approaches have been proposed to achieve stable hopping or bounding. From the perspective of theoretical analysis, Blickhan [6] proposed the idea of using a spring-mass model to describe the interdependency of mechanical parameters from which human running and hopping can be characterized. Berkemeier [7] analyzed the stability of quadrupedal bounding and pronking (hopping) through a simplified spring-damper model and proved that the linearized bound of the unperturbed model is always neutrally stable if the dimensionless body inertia is greater than 1. Ahmadi et al. [8] analyzed the limit cycles of a one-leg hopping robot and proposed an approach that would force the system trajectory to converge to the reference limit cycle. As for controlling real robots, Zabihi and Alasty [9] used fuzzy logic control to achieve a stable limit cycle for a hopping robot with two springs on the two sides. Hale et al. [10] proposed adaptive algorithms to adjust the hopping height of a one-leg robot, in which the algorithms were based on information obtained from previous hops or steps. Park et al. [1,2] proposed a variable-speed quadrupedal bounding method based on simple impulse planning. Experiments showed that the algorithm enabled the MIT Cheetah 2 to successfully achieve 3D running. Liu et al. [11] proposed a control approach by converting the time-dependent limit cycle motions into time-invariant virtual constraints; thus, the robot could be controlled to converge to a specific state of bounding. Apart from model-based methods, biologically inspired methods can offer other perspectives on efficient jumping. Khoramshahi et al. [12,13] and Buchli et al. [14,15] proposed different frequency adaptive oscillators

Received April 28, 2019; accepted July 12, 2019

Boxing WANG, Chunlin ZHOU (✉), Ziheng DUAN, Qichao ZHU,
Jun WU, Rong XIONG
State Key Laboratory of Industrial Control Technology, College of
Control Science and Engineering, Zhejiang University, Hangzhou
310027, China
E-mail: c_zhou@zju.edu.cn

Jun WU
Binhai Industrial Technology Research Institute of Zhejiang University,
Tianjin 300301, China

to automatically converge to the resonance frequency of a hopping system with springy legs, and their proposed scheme resulted in reduced energy cost. However, majority of existing proposed controllers are designed for torque-controlled robots, which are unsuitable for small-sized prototype tests. Moreover, in small-sized prototype tests, position-controlled motors are usually used.

A position-controlled robot may be able to jump by mounting elastic elements, subsequently reducing energy loss during touchdown impact. Among different methods, serial elastic actuation [16–18] has been widely used. A serial elastic actuator contains an elastic element in series and a mechanical energy source. Thus, torque control can be achieved through end-effector feedback, and then algorithms based on torque control can be transplanted. Unlike in typical serial elastic actuation, a passive spring-loaded leg mechanism with multiple segments was proposed by Spröwitz et al. [19] to increase the robustness of a quadruped robot during trotting and stepping down events. Nevertheless, regardless of whether active-control springs or passive-control springs are used, the value of the spring stiffness and the manner of mounting elastic components can affect system dynamics. A suitable control algorithm cannot be determined until compliant elements are adequately chosen and mounted.

In this study, we attempt to solve the above problems by proposing a control framework that can achieve stable quadrupedal hopping, bounding, and translational/rotational motion control on a trampoline (Fig. 1). The proposed framework, which is based on position control, does not rely on the explicit dynamic model. We choose trampoline tests because, from the perspective of energy utilization, a robot cannot possibly realize off-ground jumping if it fails to jump on a trampoline. Jumping on a trampoline requires a simplified model similar to that of regular jumping, and this similarity allows us to ignore the problem of choosing and mounting springs. Here, the control algorithms can be designed initially for the trampoline tests and then used to select and mount the suitable elastic components. To the best of our knowledge,

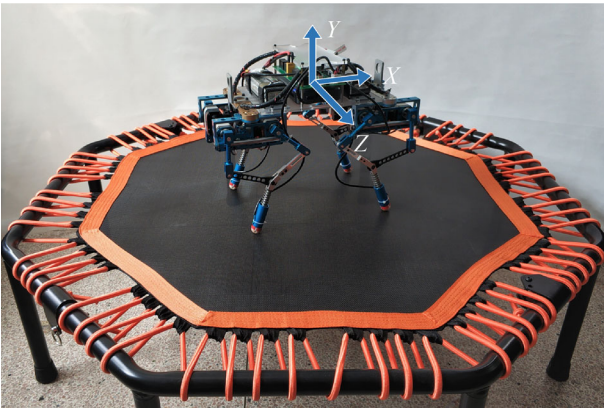


Fig. 1 Overview of the trampoline and the quadruped robot.

although some researchers have addressed the problem of balance control for robots standing on compliant surfaces [20], while other researchers have addressed the problem of stabilizing the periodic orbits of 1-degree of freedom (1-DOF) hopping robots over compliant surfaces [21], the problem of quadrupedal jumping control on a trampoline has not been addressed.

This paper is presented as follows. In Section 2, an analysis of the simplified model of jumping on a trampoline is discussed. A comparison between regular springy leg experiments and trampoline experiments is presented. In Section 3, the controller for stable hopping and bounding is introduced in detail. In Section 4, the experimental results are presented and then used to validate the proposed controller. In Section 5, drawbacks and future work are discussed.

2 Model analysis

2.1 Simplified sagittal plane model

Hopping or bounding on a trampoline can be represented as a two-legged sagittal plane model, as shown in Fig. 2(a). The robot modeled in this study is a homogeneous link with two massless legs. The origin of the coordinate is set on the surface of the trampoline, in which the x and y axes represent the horizontal and the vertical directions. (x_c, y_c) , (x_h, y_h) and (x_f, y_f) are the positions of the center of mass (CoM), the hind leg foot-end and the foreleg foot-end, respectively. θ is the pitch angle of the robot. L_b is half the length of the body. L_f and L_h represent the lengths of the foreleg and the hind leg, respectively. β_f and β_h represent the hip joint angles of the foreleg and the hind leg, respectively. For simplicity, we assume $\beta_f = \beta_h = \pi/2$ and $L_f = L_h = L$.

Ground reaction forces are used to describe the dynamics of the model.

$$m\ddot{x}_c = f_f^x + f_h^x, \quad (1)$$

$$m\ddot{y}_c = -mg + (f_f^y + f_h^y), \quad (2)$$

$$\begin{aligned} I\ddot{\theta} = & f_f^y(x_f - x_c) + f_h^y(x_h - x_c) + f_f^x(y_c - y_f) \\ & + f_h^x(y_c - y_h), \end{aligned} \quad (3)$$

where f_h^x and f_h^y represent the ground reaction forces along the x and y axes of the hind legs, f_f^x and f_f^y represent the ground reaction forces of the forelegs, I is the inertia of the body, m is the mass of the body, and g is gravity acceleration.

Assume that the dynamics of the trampoline can be modeled as separate springs. Then, $f_{f,h}^y$ can be calculated as

$$f_f^y = -k_s y_f, \quad (4)$$

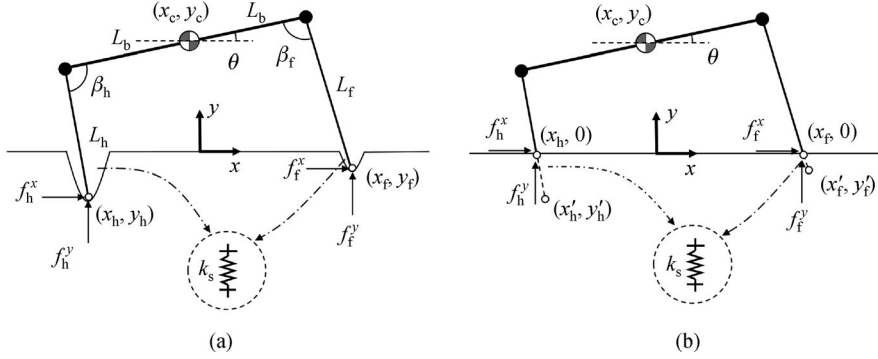


Fig. 2 (a) Simplified sagittal plane model of trampoline tests; (b) simplified model with springy legs.

$$f_h^y = -k_s y_h, \quad (5)$$

where k_s is the spring stiffness, which is a positive constant. However, for $f_{f,h}^x$, which correspond to the static friction at the toes (no slips are assumed here), they are governed by friction laws and the kinematic state of the body.

If a robot with springy legs jumps on a hard surface, as shown in Fig. 2(b), then Eqs. (1)–(3) remain to be correct, but the values of $f_{f,h}^y$ and $f_{f,h}^x$ should be

$$f_f^y = -k_s y'_f, \quad (6)$$

$$f_h^y = -k_s y'_h, \quad (7)$$

$$f_f^x = k_s (x_f - x'_f), \quad (8)$$

$$f_h^x = k_s (x_h - x'_h), \quad (9)$$

where (x'_f, y'_f) and (x'_h, y'_h) represent the rest points of the forelegs and the hind legs.

A comparison of Eqs. (6)–(9) with Eqs. (4) and (5) indicates that when a robot has a perfect vertical hopping ($\ddot{x}_c = 0$ and $x'_{f,h} = x_{f,h}$), choosing between a trampoline or springy legs will lead to the same results. Nonetheless, using a trampoline has the advantage of storing all the kinetic energy of a falling robot. As for springy legs, elastic elements must be kept strictly vertical to avoid energy loss during touchdown impact, but this scenario is difficult to achieve for legs with rotational joints. When a robot bounds on a trampoline, the difference between Eqs. (6)–(9) and Eqs. (4) and (5) indicates that the robot may not easily achieve a stable bounding pattern. A robot with springy legs bounding on a hard surface can be modeled with $f_{f,h}^x$ and $f_{f,h}^y$ to decrease the amplitude of the pitch oscillation and control the bounding period [1,2]. Stable bounding can also be automatically achieved by using passive springy legs [22–24]. However, in the trampoline tests, we found that a stable bounding gait could not be easily achieved because the robot would start to tip over after a few seconds.

2.2 Algorithms for balance control

As observed in our experiments, algorithms for balance control are necessary regardless whether the robot hops or bounds on a trampoline; that is, regardless of hopping or bounding, the body pitch angle θ could not be maintained naturally to an acceptable range. Any unsynchronized touchdown event of the forelegs and hind legs would induce pitch oscillation due to the springiness of the surface. Without balance control, the robot would either tip over or move chaotically.

In terms of hopping, $f_{f,h}^y$ is effective in suppressing the oscillation of θ during the stance phase. The corresponding control law is

$$[\Delta y_f \quad \Delta y_h]^T = [-1 \quad 1]^T k_{py} \theta, \quad (10)$$

where Δy_f and Δy_h are the output positional increments for the forelegs and the hind legs, and k_{py} is a positive constant. According to Eqs. (4) and (5), positional increments will affect $f_{f,h}^y$. Thus, a torque can be applied to maintain the value of θ .

Equation (10) differs from the common negative-feedback control law and can offset θ . By taking Fig. 2(a) as an example (where $\theta > 0$), the common θ offset law becomes $[\Delta y_f \quad \Delta y_h]^T = [1 \quad -1]^T k_{py} \theta$. This control law attempts to reduce the pitch angle θ by decreasing the length of hind legs and increasing the length of forelegs, which is simply based on kinematic relations. According to Eqs. (3)–(5), assuming $\sin \theta \approx \theta$, the resultant dynamics of θ is

$$I\ddot{\theta} = -2k_s(y_c L + L_b^2 - L^2)\theta - 2k_{py}k_s L\theta + D, \quad (11)$$

where $D = f_f^x(y_c - y_f) + f_h^x(y_c - y_h)$. However, Eq. (10) entails a strategy opposite to the common θ offset law. The resultant dynamics can be expressed as

$$I\ddot{\theta} = -2k_s(y_c L + L_b^2 - L^2)\theta + 2k_{py}k_s L\theta + D. \quad (12)$$

Usually, $L_b > L$ and $f_{f,h}^x$ have sufficiently small values and can be ignored in hopping. Thus,

$-2k_s(y_c L + L_b^2 - L^2) < 0$ and $D \approx 0$. Moreover, without the balance control ($k_{py} = 0$ in Eq. (11)), the trampoline will turn any disturbance of θ into an oscillation. A comparison of Eq. (11) with Eq. (12) indicates that the common negative-feedback control law can increase the amplitude of the oscillation, a scenario that hinders the stabilization of the robot. Equation (10) can then be used to reduce the amplitude of the oscillation by neutralizing the item $-2k_s(y_c L + L_b^2 - L^2)\theta$; in this manner, the amplitude of the oscillation can be contained.

In terms of bounding, a control law similar to Eq. (10) is used to stabilize the robot during the stance phase.

$$\begin{cases} [\Delta\dot{x}_f, \Delta\dot{y}_f] = [-k_{bx}, k_{by}] \cdot |\dot{\theta}|, & y_f < 0, \dot{\theta} < 0, \\ [\Delta\dot{x}_f, \Delta\dot{y}_f] = [k_{bx}, k_{by}] \cdot |\dot{\theta}|, & y_h < 0, \dot{\theta} > 0, \end{cases} \quad (13)$$

where k_{px} and k_{py} are positive constants. Output positional increments can only be applied to the supporting legs. Equation (13), unlike Eq. (10), is designed to more radically reduce $f_{f,h}^y$ and $f_{f,h}^x$ caused by the trampoline. First, the equation uses $\dot{\theta}$ instead of θ to generate the output. Thus, the robot can react much more quickly. Second, the use of $\dot{\theta}$, $\Delta\dot{x}_f$, and $\Delta\dot{y}_f$ enables the affected leg to hold its length even when the pitch angle returns to zero or stops increasing. Given that $\Delta\dot{x}_f$ will reduce the relative movement trends between the foot-end and the trampoline, $f_{f,h}^x$ can be reduced by Eq. (13). The experiments show that Eq. (13) is more effective than Eq. (10) in stabilizing the bounding robot.

2.3 Effects of leg stretch delay on hopping height

The robot will lose its mechanical energy at each jumping period due to the touchdown impact and the damping characteristics of a real trampoline. Thus, the supporting legs will stretch at a specific time instant during the stance phase to sustain a stable jumping pattern. This time instant is expressed as the time delay from the touchdown event, and it is called leg stretch delay in this study.

Given a certain leg stretch delay, the hopping height will naturally converge to a steady-state value that can be illustrated by a simplified model. As shown in Fig. 3(a), the hopping robot in the model can be simplified as a mass block with a single stretchable leg, and energy loss can be described by the velocity scaling factor k_v during touchdown events. The origin of the coordinate is set on the surface of the trampoline. The dynamics of this model is

$$\begin{cases} m\ddot{y}_c = -mg + f^y, \\ f^y = \max(-k_s(y_c + l_c), 0), \\ \dot{y}_c(t_*^+) = k_v \dot{y}_c(t_*), \end{cases} \quad (14)$$

where l_c is the leg length, t_* is the time instant when the robot touches the compliant surface, k_v is a positive number with an absolute value not larger than 1, and f^y is the foot contact force.

Figure 3(b) illustrates a single hopping cycle of the simplified model, as described in Eq. (14), in which the leg stretch event is not considered (l_c is kept to zero). In this study, hopping cycle is defined as the period between the time instant of $\dot{y}_c = 0, y_c > 0$ to the next instant. The dashed line in Fig. 3(b) is taken as an example. Hopping cycle starts from $\dot{y}_c = 0, y_c > 0$ to $\dot{y}_c = 0, y_c < 0$ and then back to $\dot{y}_c = 0, y_c > 0$. When $k_v = 1$, as shown by the dashed line, no energy is lost at touchdown impact. The trajectory is a closed curve, which means that the robot can reach its initial vertical position after hopping on a compliant surface. However, when $k_v = 0.8$, \dot{y}_c will be reduced to $0.8\dot{y}_c$ at the touchdown impact ($y_c = 0$). As shown by the solid line, the robot will not be able to reach its initial position due to energy loss.

By using \bar{y}_c to represent the initial point of a hopping cycle (also the highest point reached by the robot in the hopping cycle), the robot vertical position $y_c(t)$ during the stance phase can be solved by Eq. (14):

$$y_c(t) = -k_v \sqrt{\frac{2mg\bar{y}_c}{k_s}} \sin \sqrt{\frac{k_s}{m}} t + \frac{mg}{k_s} \cos \sqrt{\frac{k_s}{m}} t - \frac{mg}{k_s}, \quad (15)$$

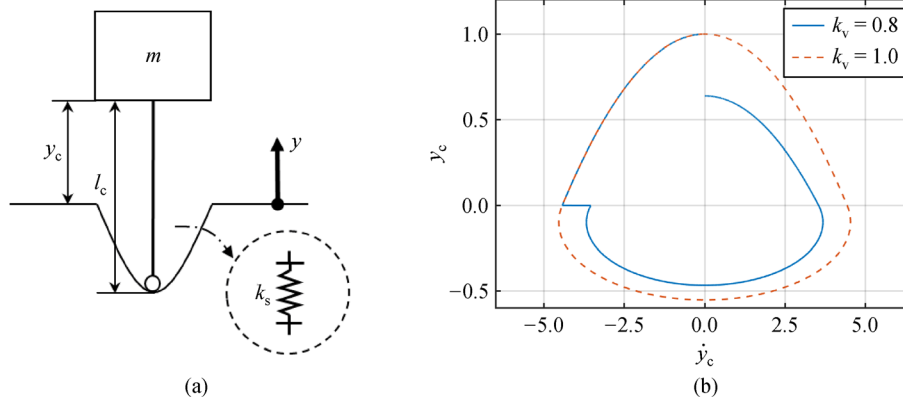


Fig. 3 (a) Simplified model of a hopping robot; (b) one hopping cycle.

where t starts from the moment when the robot touches the compliant surface. Leg stretch is not considered in Eq. (15).

Then, by using t_d to represent the preset time delay, the spring compression (which is a non-negative length variable) at t_d is

$$\hat{y}_s = -y_c(t_d), \quad (16)$$

where \hat{y}_s is the spring compression at t_d .

For simplicity, we assume that the leg stretch is completed immediately at t_d , where l_c is changed from zero to $-\Delta l$ ($\Delta l > 0$). Then, the energy injected by the leg stretch is

$$\Delta E^\uparrow = \frac{1}{2}k_s[\hat{y}_s + \Delta l]^2 - \frac{1}{2}k_s\hat{y}_s^2 = \frac{1}{2}k_s(\Delta l)^2 + k_s\Delta l\hat{y}_s. \quad (17)$$

Energy loss during touchdown impact is

$$\Delta E^\downarrow = \frac{1}{2}m\dot{y}_c^2(t_*^-) - \frac{1}{2}m\dot{y}_c^2(t_*^+) = mg\bar{y}_c(1 - k_v^2). \quad (18)$$

Thus, in the next hopping cycle, the highest point that can be reached by the robot is

$$\bar{y}_c[N+1] = \bar{y}_c[N] + \frac{\Delta E^\uparrow - \Delta E^\downarrow}{mg}, \quad (19)$$

where $\bar{y}_c[N]$ represents the highest jumping point of the N th hopping cycle, and $\bar{y}_c[N+1]$ represents the one of the next hopping cycle.

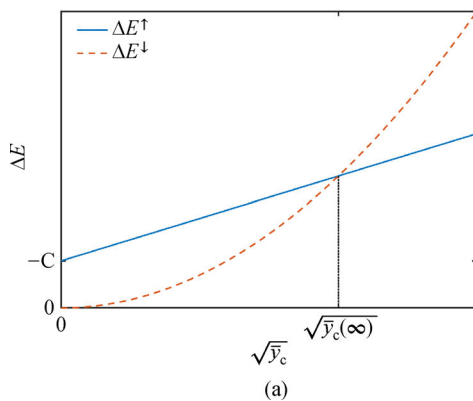
As indicated by Eq. (19), \bar{y}_c will converge to a stable value $\bar{y}_c(\infty)$ when $\Delta E^\uparrow = \Delta E^\downarrow$. The solution of $\Delta E^\uparrow = \Delta E^\downarrow$ is

$$\bar{y}_c(\infty) = \left(\frac{-B + \sqrt{B^2 - 4AC}}{2A} \right)^2, \quad (20)$$

where

$$A = mg(1 - k_v^2),$$

$$B = -k_s k_v \Delta l \sqrt{2mg/k_s} \sin(\omega_n t_d),$$



$$C = k_s(\Delta l)^2/2 - [1 - \cos(\omega_n t_d)]mg\Delta l,$$

$$\omega_n = \sqrt{k_s/m}.$$

The definitions imply that $B^2 - 4AC > B^2 > 0$ because $A > 0$ and $C < 0$. Thus, a reasonable solution will always exist for $\Delta E^\uparrow = \Delta E^\downarrow$.

Figure 4(a) is plotted to explain comprehensively the convergence process of \bar{y}_c . According to Eqs. (17) and (18), ΔE^\uparrow is a line function while ΔE^\downarrow is a quadratic function with respect to $\sqrt{\bar{y}_c}$. Both equations have an intersection point at $\sqrt{\bar{y}_c} = \sqrt{\bar{y}_c(\infty)}$, as indicated by Eq. (20). In Fig. 4(a), if $\sqrt{\bar{y}_c} < \sqrt{\bar{y}_c(\infty)}$, then $\Delta E^\uparrow > \Delta E^\downarrow$; therefore, $\sqrt{\bar{y}_c}$ increases according to Eq. (19). If $\sqrt{\bar{y}_c} > \sqrt{\bar{y}_c(\infty)}$, then $\Delta E^\uparrow < \Delta E^\downarrow$; therefore, $\sqrt{\bar{y}_c}$ decreases. Moreover, $\sqrt{\bar{y}_c}$ will converge toward $\sqrt{\bar{y}_c(\infty)}$. Figure 4(b) illustrates the evaluation of \dot{y}_c and y_c with the leg stretch enabled and with k_v set to a value smaller than 1. The values of parameters are as follows: $m = 5$, $k_s = 5000$, $k_v = 0.8$, $\Delta l = 0.02$, $\bar{y}_c[1] = 1$, and $t_d = 0.1\sqrt{k_s/m} \approx 0.020$. As illustrated by Fig. 4(b), the trajectory becomes much denser when the hopping cycle increases. Thus, the trajectory tends to form a closed curve, in which \bar{y}_c is unchanged.

In conclusion, \bar{y}_c will always converge to a single $\bar{y}_c(\infty)$. Thus, stabilizing the hopping height \bar{y}_c by adjusting leg stretch length or by using other parameters is unnecessary, and \bar{y}_c will reach $\bar{y}_c(\infty)$ naturally after sometime.

Nevertheless, the stable value of \bar{y}_c can be adjusted with Δl and t_d . According to Eq. (20), Δl is positively correlated with $\bar{y}_c(\infty)$. However, the relation between t_d and the stable value of $\bar{y}_c(\infty)$ is somewhat complex. Figure 5(a) illustrates how Δl and t_d affect $\bar{y}_c(\infty)$. The values of the other parameters are as follows: $m = 5$, $k_s = 5000$, and $k_v = 0.7$. $\bar{y}_c(\infty)$ increases as Δl increases when a specific value of t_d is selected. However, when Δl is fixed, t_d can be used to ensure the largest value of $\bar{y}_c(\infty)$. Figure 5(b) presents the relation between $\bar{y}_c(\infty)$ and t_d with different values of k_s . The values of the other parameters are as

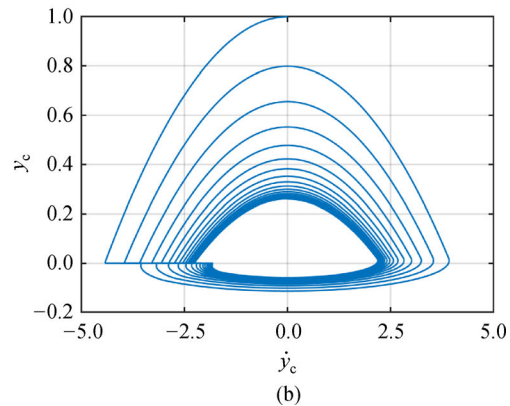


Fig. 4 (a) \bar{y}_c convergence in $\Delta E - \sqrt{\bar{y}_c}$ plot; (b) trajectory convergence in $\dot{y}_c - y_c$ plot.

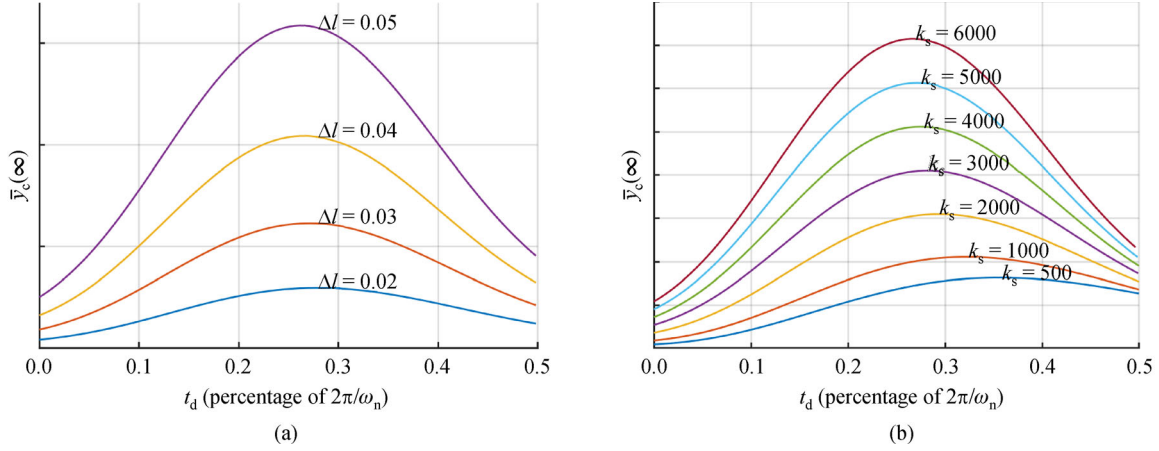


Fig. 5 (a) Different values of t_d and Δl affect $\bar{y}_c(\infty)$; (b) different values of k_s affect the t_d value when deciding for the largest $\bar{y}_c(\infty)$.

follows: $m = 5$, $\Delta l = 0.03$, and $k_v = 0.7$. In the figure, the t_d values for the corresponding largest $\bar{y}_c(\infty)$ values are in the range of 20% to 40% of $2\pi/\omega_n$, and they move close to 25% of $2\pi/\omega_n$ when k_s increases. This phenomenon can be explained by the stance phase, which consists of the following three parts: 1) The robot decelerates to the balance point where the spring force equals the robot weight; 2) the robot reaches the lowest point and then reverts to the balance point; and 3) the robot decelerates until it leaves the compliant surface. The time of completing the second part is constant, which is 50% of $2\pi/\omega_n$. However, the time needed for the first and third parts depends on contact velocity and the values of k_s and m . If the time consumed by the first and third parts is relatively short, then $t_d = 0.25 \cdot 2\pi/\omega_n$ leads \hat{y}_s to move close to its maximum value. Therefore, according to Eq. (17), this scheme enables ΔE^\dagger to reach its maximum value, thus leading to the largest $\bar{y}_c(\infty)$. Increasing the values of k_s shortens the time consumed by the first and third parts; thus, t_d associated with the largest $\bar{y}_c(\infty)$ is at approximately 25% of $2\pi/\omega_n$.

The plot in Fig. 5(b) also indicates that a larger k_s would result in a larger $\bar{y}_c(\infty)$. The figure is plotted with the same amplitude of Δl . Thus, according to Eq. (17), the energy injected into the system grows as k_s increases. Therefore, the largest value of $\bar{y}_c(\infty)$ also increases. For researchers who can set the value of k_s in real experiments, k_s can be used to adjust the largest value of $\bar{y}_c(\infty)$. However, a large k_s will reduce the value of the leg stretch delay to reach the largest $\bar{y}_c(\infty)$. That means more responsive actuators are required.

3 Controller structure

The proposed controller consists of the following three parts: Balance-keeping motion generator, translational and rotational motion generator, and basic-jumping pattern generator. These three types of motions are added to derive

the final foot-end positions. We choose this parallel motion generation structure because it is easy to deploy for online multitask switching while keeping the robot balanced [25,26].

Figure 6 illustrates the general structure of the proposed controller. θ is the pitch angle of the robot. TD_i represents a touchdown event. LO_i represents a liftoff event. Matrix \mathbf{P} is the controller output, which contains four 3D foot-end position vectors \mathbf{p}_i with respect to the body coordinate (Fig. 1). The index i represents the leg number, including LF (left foreleg), RF (right foreleg), LH (left hind leg), and RH (right hind leg).

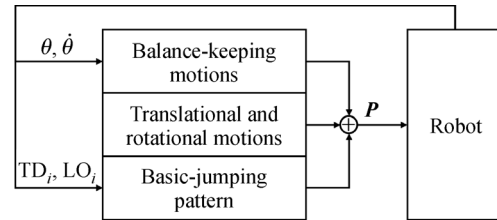


Fig. 6 Proposed controller structure.

The basic-jumping pattern generator initiates primitive leg motions to achieve hopping and bounding, and disturbances are not induced. In this condition, legs can simply stretch when they are on the ground, and they contract when they are off the ground. The motions generated by this module are defined as

$$\dot{\phi} = T/(2\pi), \quad \phi \in [0, 2\pi], \quad (21)$$

$$y_d = \begin{cases} \frac{l_A}{2} \sin\left(\omega_1 \phi - \frac{\pi}{2}\right) + \frac{l_A}{2} & 0 < \phi \leq \phi_{co}; \\ l_A & \phi_{co} < \phi \leq \pi; \\ \frac{l_A}{2} \sin\left[\omega_2(\phi - \pi) + \frac{\pi}{2}\right] + \frac{l_A}{2} & \pi < \phi \leq \phi_{st}; \\ 0 & \phi_{st} < \phi \leq 2\pi. \end{cases} \quad (22)$$

$$\omega_1 = \pi/\phi_{co}, \quad \omega_2 = \pi/(\phi_{st} - \pi),$$

where y_d is the output 1D trajectory along the vertical direction, l_A is the amplitude of the trajectory and is used to define the leg stretch amplitude, which corresponds to Δl in Section 2, T is the trajectory period, ϕ_{co} is the endpoint of the leg-contracting process, and ϕ_{st} is the endpoint of the leg-stretching process. Equations (21) and (22) form a simple phase oscillator that generates periodic signals. (See Refs. [27,28] for additional information on oscillators with arbitrarily defined limit cycles.)

As shown in Fig. 7, within one period, the leg first contracts with amplitude l_A and then holds the position until $\phi = \pi$. When $\pi < \phi < \phi_{st}$, the leg stretches and keeps its maximum stretch length to the end. Each of the four legs is assigned such a predefined trajectory.

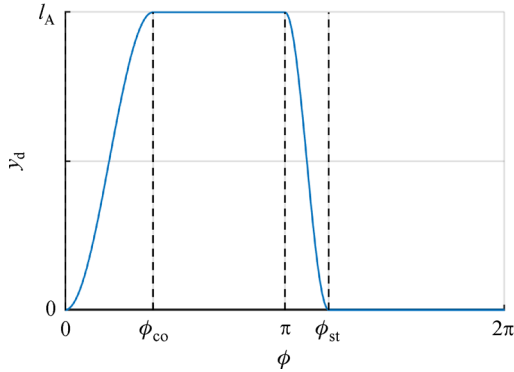


Fig. 7 Predefined basic-pattern pattern.

In hopping, touchdown and liftoff events are induced to reset ϕ and enable the predefined period to adapt to the real jumping period.

$$\phi_i = \begin{cases} \pi - \phi_{off} & \text{TD}_{all} = \text{true;} \\ 0 & \text{LO}_{all} = \text{true.} \end{cases} \quad (23)$$

where ϕ_{off} , which equals $2\pi t_d/T$, is set to delay the leg stretch after the TD event; this variable has a predefined constant value. TD_{all} and LO_{all} are used to synchronize the stretches and contractions of the four legs. TD_{all} or LO_{all} represents the event in which all four legs have touched or left the ground. As illustrated in Fig. 5, the value of ϕ_{off} or t_d will affect $\bar{y}_c(\infty)$. However, controlling the hopping height is not one of the objectives of this study. Thus, the value of ϕ_{off} is chosen through trial and error to attain the most stable hopping state. Nonetheless, we also present the results of the different hopping heights with varying ϕ_{off} in the experiment section.

In bounding, the forelegs and hind legs alternatively support the robot. The phase reset law is

$$\phi_{f,h} = \begin{cases} \pi & \text{TD}_{f,h} = \text{true, } \dot{\theta} \in [\dot{\theta}_l, \dot{\theta}_u], \\ 0 & \text{LO}_{f,h} = \text{true,} \end{cases} \quad (24)$$

where $\dot{\theta}_l$ and $\dot{\theta}_u$ are the lower and upper limits of the triggered $\dot{\theta}$, respectively. These variables have small values (near zero). As the pitch angle oscillates at a specific period in the bounding gait, Eq. (24) can be used to control or stretch the supporting forelegs or hind legs without explicitly affecting the amplitude of the pitch oscillation. The subscript “f,h” indicates that forelegs and hind legs are controlled in pairs. ϕ_{off} is no longer considered because $\dot{\theta}$ already represents the triggered moment.

Translational and rotational motions are generated through simple kinematic laws:

$$\dot{\mathbf{p}}_t = [-v_{dx}, 0, -v_{dz}]^T, \quad (25)$$

$$\dot{\mathbf{p}}_r = -\Omega_d \times \mathbf{p}_t, \quad (26)$$

where \mathbf{p}_t and \mathbf{p}_r are the output foot-end positional increments of the translational and rotational motions, respectively, v_{dx} is the desired body velocity along the heading direction, v_{dz} is the desired velocity along the lateral direction, and Ω_d is the desired body angular velocity.

Furthermore, for the fixed set point control for the yaw angle, the following equation is used:

$$\dot{\mathbf{p}}_r = k_{yaw}(\psi_m - \psi_d) \cdot [0, 1, 0]^T \times \mathbf{p}_t, \quad (27)$$

where ψ_m and ψ_d are the measured and desired body yaw angles, respectively, and k_{yaw} is a tunable positive parameter.

Balance-keeping motions are generated according to Eqs. (10) and (13). For hopping,

$$\mathbf{p}_b = \begin{cases} [0, -1, 0]^T k_{py} \theta & \text{for forelegs,} \\ [0, 1, 0]^T k_{py} \theta & \text{for hindlegs.} \end{cases} \quad (28)$$

For bounding,

$$\dot{\mathbf{p}}_b = \begin{cases} [-k_{bx}, k_{by}, 0]^T |\dot{\theta}| & \text{for forelegs,} \\ [k_{bx}, k_{by}, 0]^T |\dot{\theta}| & \text{for hindlegs,} \end{cases} \quad (29)$$

where $\dot{\theta}$ is coerced into $[-\infty, 0]$ for the forelegs and $[0, +\infty]$ for the hind legs.

Equations (25)–(29) are only applied to supporting legs, their values will be cleared when the corresponding leg enters the swing phase.

4 Experiments

Experiments are performed on a untethered servo-actuated quadruped robot and a commercial adult-used trampoline (Fig. 1). The quadruped robot (Fig. 8) weighs approximately 4.5 kg and has a length of 450 mm, width of 180 mm, and height of 270 mm. Each of its legs has three servos that can drive the foot-end to move in the sagittal

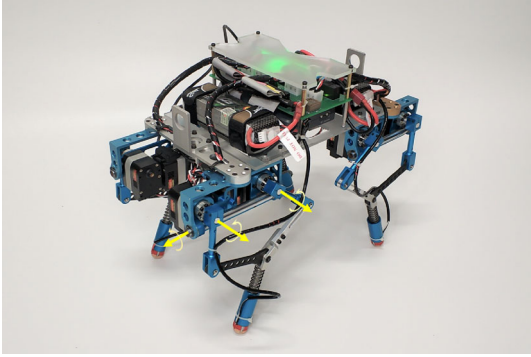


Fig. 8 Servo-actuated quadruped robot.

plane and rotate in the horizontal axis. The three servo axes are depicted by arrows in Fig. 8. 3D foot-end position control can be achieved within the workspace. Four tactile switches are mounted on the foot-ends to detect touchdown and liftoff events. An onboard inertial measurement unit (IMU) is used to detect the attitude angle of the robot. As for the trampoline, the diameter of the jumping cloth is approximately 92 cm. Elastic ropes are used to tie the jumping cloth to the outer metal frame. An overview of the successful hopping and bounding is shown in Fig. 9. The experimental video of the performed tests can be found from the Electronic Supplementary Material.

4.1 Hopping

The parameters used in the hopping tests are listed in Table 1. Notably, T is not the period of real hopping. According to Eq. (23), the predefined period T will be adjusted automatically when T is greater than the actual period. The robot can automatically start the hopping without being dropped from a specific height. Because the predefined trajectory (Fig. 7) will make the leg contract first, and this scheme is faster than free-fall dropping. Thus, the initial height equals l_A , and it will converge to a stable value, as illustrated in Fig. 4(a). Moreover, l_A is set to be as large as possible to increase the maximum jumping height. However, given that the vertical workspace of the robot is limited to 55 mm, the value of 30 mm can be chosen to leave enough margin for balance-keeping motions.

Table 1 Control parameters in the hopping tests

T/s	ϕ_{co}	ϕ_{st}	ϕ_{off}	l_A/mm	$k_{py}/(mm \cdot ^\circ)^{-1}$	k_{yaw}
2	0.08π	0.035π	0.025π	30	5	0.08

Figure 10(a) presents the recorded pitch angle data of successful (solid line) and failed (dashed line) hopping. In the failed hopping test, in which the balance-keeping motion generator was turned off, the pitch angle rapidly accumulated. After two or three jumps, the pitch angle became excessively large that the robot dropped onto the trampoline only with two forelegs or hind legs, while the other side fell due to gravity. Thus, as shown by the dashed line in the figure, stable periodic hopping cannot be achieved without balance-keeping motions.

As a mean to fully understand successful hopping, Fig. 10(b) is presented to show the stable pattern achieved in this study. The dashed square wave in Fig. 10(b) represents the periods when at least three foot-ends have touched the trampoline. The periodic appearance of the higher value of the square wave indicates that the hopping motion has reached a stable state. In our experiments, the time in which the robot stayed on the ground was approximately 0.10 s, and the time in which the robot stayed on air was approximately 0.25 s.

Rotational and translational motions induced disturbances to the pitch angle. Within the proposed control framework, the translational or rotational velocity was limited to a small value to guarantee stability. For lateral motion, the desired velocity v_{dz} was set to 0.6 m/s. For forward or backward motion, the desired velocity v_{dx} was set to 0.15 m/s. As an example, the recorded attitude angle data with rotation motions are obtained, as shown in Fig. 11. Equation (27) was used, but the integration period (100 ms) was set to 20 times of the control period (5 ms) to ensure that the over-frequent yaw angle adjustments could be avoided. From 38 to 42 s, as shown by the dashed line, rotational motions were added, and a 45° turn was successfully achieved. Disturbances were observed in both pitch and roll angles, but they diminished after the rotational motions were conducted. Thus, the proposed controller could stabilize the robot after limited interference.

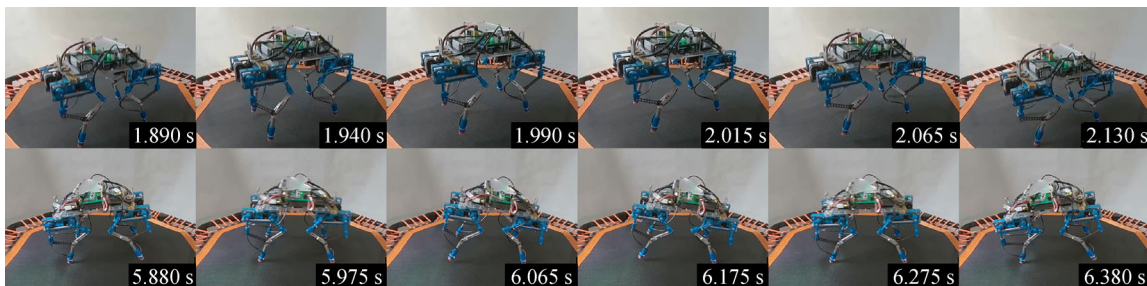


Fig. 9 Overview of successful hopping (above) and bounding (bottom).

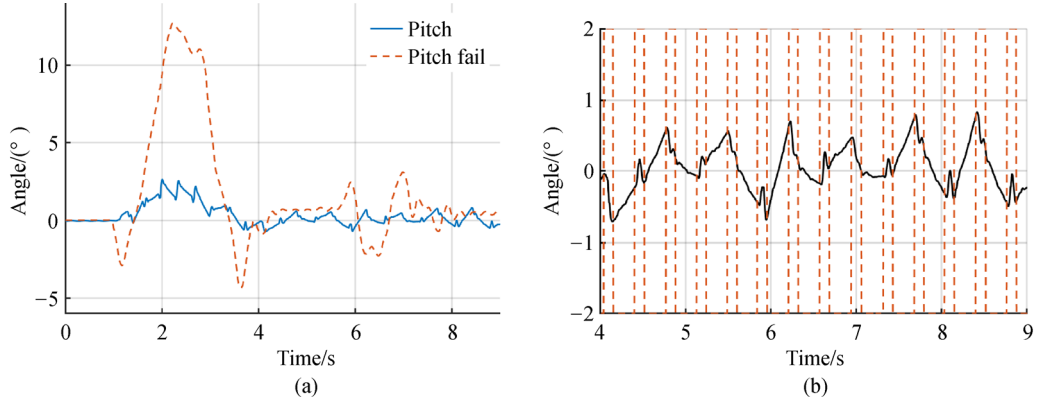


Fig. 10 (a) Recorded pitch angle data of successful and failed hopping tests; (b) recorded pitch angle data and on-ground indicators.

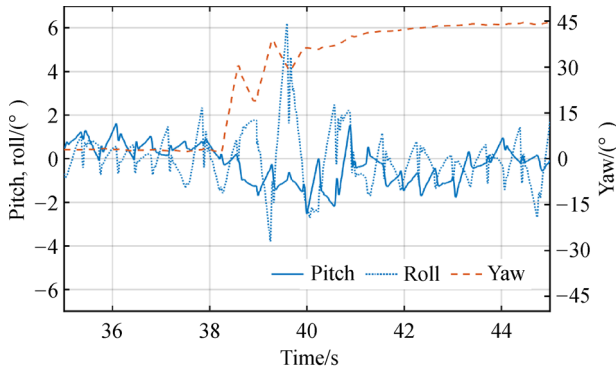


Fig. 11 Recorded attitude data with rotational motions during hopping.

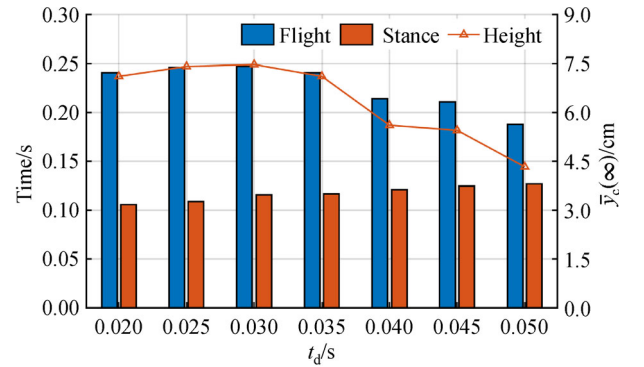


Fig. 12 Recorded flight and stance phase durations with different t_d .

Unlike the pitch angle, the roll angle could be naturally stabilized into a small range near zero. The difference could be attributed to the ratio of the body length and the body width. As indicated in Eq. (12), when the roll angle is considered, $2L_b$ should correspond to the width rather than the length of the robot. Normally, the body width is much smaller than the length. Thus, the dynamics could be changed. However, a comprehensive exploration of this problem is beyond the scope of this study.

The recorded flight and stance phase durations with different t_d is plotted to verify the relation between t_d and $\bar{y}_c(\infty)$, as shown in Fig. 12. The values of the other parameters are as follows: $T = 2$ s, $\phi_{co} = 0.08\pi$, $\phi_{st} = 0.02\pi$, and $A = 30$ mm. At a specific value of t_d , the flight or stance phase duration fluctuated, even after dozens of hopping cycles; hence, we calculated the mean value within a time period, particularly when the fluctuation was relatively small. $\bar{y}_c(\infty)$ was derived by considering the flight phase duration and by assuming a robot with perfect free-fall movements. As shown in Fig. 12, a maximum $\bar{y}_c(\infty)$ of 7.5 cm could be achieved at approximately $t_d = 0.03$ s. Other values of t_d would lead to a much smaller $\bar{y}_c(\infty)$. When $t_d > 0.04$ s, $\bar{y}_c(\infty)$ was drastically reduced. Unlike the unit in Fig. 5, the horizontal axis unit

of Fig. 12 falls second, but the value is not presented as the percentage of $2\pi/\omega_n$ because ω_n cannot be precisely estimated in our experiments. However, given that the stance phase duration is mainly composed of π/ω_n , the short stance phase duration of 0.106 s may be used as a rough estimation for π/ω_n . Thus, on the basis of Fig. 5, the largest $\bar{y}_c(\infty)$ can be achieved at approximately 25% of $2\pi/\omega_n$, but this finding conflicts with the result presented in Fig. 12. In particular, in Fig. 12, the largest $\bar{y}_c(\infty)$ was achieved at approximately 0.03 s, which was only 14% of the rough estimation of $2\pi/\omega_n$. This deviation may be explained by the robot needing some time to stretch its leg, whereas in mathematical analysis, the leg stretch can be completed immediately. In our test, ϕ_{st} was set to 0.02π in that the predefined leg stretch should be finished within 0.02 s. If this time is considered, then the “real” t_d should be approximately 0.05 s, which is approximately 25% of the rough estimation of $2\pi/\omega_n$. Thus, the experiment results match the theoretical prediction.

4.2 Bounding

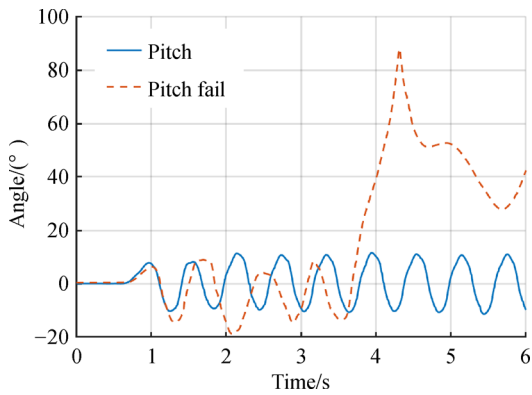
The parameters of the bounding tests are listed in Table 2. We also set a range limit to the final \mathbf{p}_b output in Eq. (29);

Table 2 Control parameters in the bounding tests

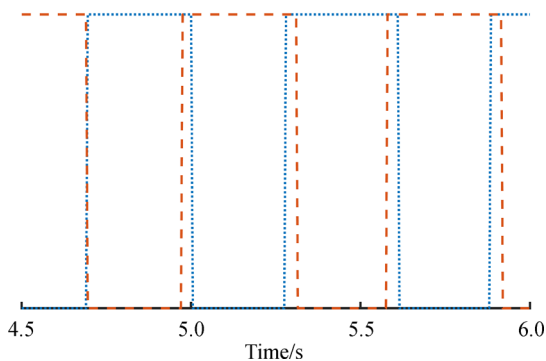
T/s	ϕ_{co}	ϕ_{st}	l_A/mm	$k_{by}/(m \cdot s^{-1} \cdot (^\circ)^{-1} \cdot s)$	$k_{bx}/(m \cdot s^{-1} \cdot (^\circ)^{-1} \cdot s)$	$\dot{\theta}_l/(^\circ \cdot s^{-1})$	$\dot{\theta}_u/(^\circ \cdot s^{-1})$	k_{yaw}
2	0.08π	0.035π	30	0.04	0.04	-30	-10	0.08

in this manner, the leg stretch will not be completely ignored. For the above parameters, the range limit is $[-15, 15]$ mm.

Figure 13 presents the recorded pitch angle data in successful (solid line) and failed (dashed line) bounding. For the proposed controller, the bounding must start with a side lift. As observed in our experiments, a passively stable bounding on the trampoline was not easy to achieve. The robot either stopped swinging quickly (with a small value of l_A) or tipped over after a few seconds (with a big value of l_A). The dashed line in Fig. 13 provides an example of the tip-over situation when no balance-keeping motions are induced.

**Fig. 13** Recorded pitch angle data during bounding.

With the help of balance-keeping motions, the pitch angle can be stabilized into periodic oscillation, as shown by the solid line in Fig. 13. However, unlike in stable hopping, the moments in which all four legs are on air have not been observed in stable bounding. The two square waves in Fig. 14 represent the periods when the forelegs or

**Fig. 14** Alternative change of supporting legs in stable bounding.

hind legs have supported the robot. The squares spread over the timeline without any gaps; thus, a flight phase did not occur in stable bounding. The overlaid parts of the two square waves indicate that moments have occurred when all four legs supported the robot. In our experiments, these moments lasted approximately 0.04 s, and the bounding period was approximately 0.60 s.

According to the literature [20–22], stable bounding at moments in which four legs are on air can be naturally achieved. However, in our experiments, extra methods (Eq. (13)) were initially needed to stabilize the bounding gait; additionally, stable bounding at moments in which four legs are on air was impossible to achieve. As depicted by our sagittal plane models, the problem can be explained by the different horizontal contact force generating mechanisms between the two types of jumping. For jumping with springy legs, these legs could provide elastic forces along both vertical and horizontal directions. The horizontal elastic forces generated by the legs could reduce the maximum horizontal contact forces. However, for jumping with stiff legs on a trampoline, the trampoline could only generate vertical elastic forces. Thus, the horizontal contact forces were governed by friction laws and the relative motion trend between the robot and the contact surface. The resultant horizontal contact forces could be much larger and result in a much easier tipping over of the robot. In other words, dynamic bounding gait at moments in which four legs are on air is hard to achieve when jumping on a trampoline.

Figure 15 illustrates the recorded altitude angle of the lateral and rotational motion tests. In these tests, Eq. (27) was used to control the yaw angle. Similar to the hopping tests, we set the integration period (100 ms) of Eq. (27) to be 20 times of the control period (5 ms). In this manner, over-frequent yaw adjustments could be avoided. In the present lateral test result (Fig. 15(a)), we set the desired lateral velocity v_{dx} to be 0.4 m/s from 75 to 78 s after the robot entered a stable bounding state. The amplitude of the pitch oscillation slightly decreased, the frequency of the roll oscillation increased, but the amplitude stayed nearly the same. Thus, the stability of the robot was proven during lateral motion. The yaw angle drifted from -7° to -3° . This drift was in line with expectations, considering that the value of k_{yaw} was tuned to allow a small range of yaw angle deviation (precise control of the yaw angle was not needed). As for forward or backward movement, the maximum value of v_{dx} could be set to 0.2 m/s. Similar to the lateral motion tests, the changes in the amplitude or frequency of the pitch or roll angle oscillation were not apparent.

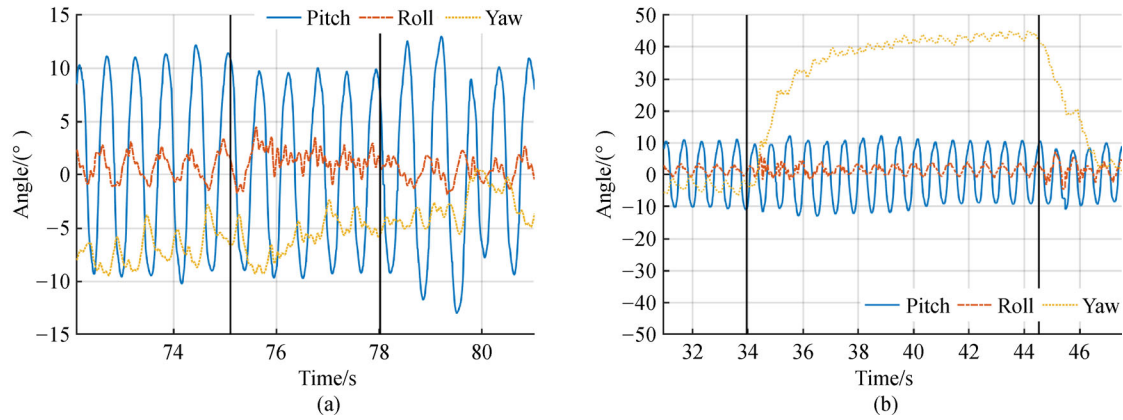


Fig. 15 Recorded attitude angle of bounding test. Recorded data of (a) lateral motion tests and (b) rotational motion tests.

In the present rotational motion test result (Fig. 15(b)), we set the desired yaw angle from 0° to 45° between 34 and 44.5 s and then reverted to 0° after 44.5 s. The yaw angle could converge to the newly desired value within 5 s. During the converging process, the amplitude and frequency of the pitch and roll angle oscillation did not show obvious differences compared with those in the no-rotation motion state. This result differed from the findings generated by the hopping tests in which large disturbances appeared in the pitch and roll angle during rotational motions (Fig. 11). By contrast, in the bounding tests, disturbances were generated when the robot started to rotate. As shown in Fig. 15(b), perturbations of the pitch and roll angle were observed at 34 and 44.5 s, particularly when the robot started rotational motion. According to Eq. (27), the perturbations can be attributed to the maximum difference between the measured and desired yaw angles when a new desired yaw angle was eventually set. At that moment, the resultant angular velocity is also at its maximum.

5 Conclusions

In this study, we proposed a control framework to achieve stable quadrupedal hopping and bounding on a trampoline. An intuitive approach based on contact-force balance was used to stabilize both the hopping and the bounding. Together with other parallel modules, the robot could perform translation and rotation motions. The control framework did not require any calculation of complex dynamics, indicating its suitability for small servo-driven quadruped robots. However, ignoring the dynamics may limit the performance of the robot and reduce its final translation and rotational speed. In the future, we plan to mount suitable elastic parts to a robot to further test the proposed control framework on hard surfaces with springy legs. For bounding in particular, detailed model analysis and comparative experiments shall be conducted.

Acknowledgements Financial support was provided by the Zhejiang Provincial Natural Science Foundation (Grant No. Y18F030012), the Science and Technology Project of Zhejiang Province (Grant No. 2019C01043), the National Natural Science Foundation of China (Grant No. 61836015), and the State Key Laboratory of Industrial Control Technology (ICT1807).

Electronic Supplementary Material The supplementary material can be found in the online version of this article (<https://doi.org/10.1007/s11465-019-0559-5>) and is accessible to authorized users.

Open Access This article is licensed under Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution, and reproduction in any medium or format, as long as appropriate credit is given to the original author(s) and the source, a link is provided to the Creative Commons license, and any changes made are indicated.

Images or other third-party materials in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Park H W, Wensing P M, Kim S. High-speed bounding with the MIT Cheetah 2: Control design and experiments. *International Journal of Robotics Research*, 2017, 36(2): 167–192
2. Park H W, Park S, Kim S. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D running of MIT Cheetah 2. In: *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, 2015, 5163–5170
3. Park H W, Wensing P M, Kim S. Online planning for autonomous running jumps over obstacles in high-speed quadrupeds. In: *Proceedings of Robotics: Science and System Conference*. Rome, 2015
4. Gehring C, Coros S, Hutter M, et al. Towards automatic discovery of agile gaits for quadrupedal robots. In: *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, 2014, 4243–4248

5. Hutter M, Gehring C, Lauber A, et al. ANYmal-toward legged robots for harsh environments. *Advanced Robotics*, 2017, 31(17): 918–931
6. Blickhan R. The spring-mass model for running and hopping. *Journal of Biomechanics*, 1989, 22(11–12): 1217–1227
7. Berkemeier M D. Modeling the dynamics of quadrupedal running. *International Journal of Robotics Research*, 1998, 17(9): 971–985
8. Ahmadi M, Michalska H, Buehler M. Control and stability analysis of limit cycles in a hopping robot. *IEEE Transactions on Robotics*, 2007, 23(3): 553–563
9. Zabihi M, Alasty A. Modeling and fuzzy control of one-legged somersaulting robot. In: *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Madrid: IEEE, 2018, 2701–2706
10. Hale M F, Du Bois J L, Irvani P. Agile and adaptive hopping height control for a pneumatic robot. In: *Proceedings of 2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane: IEEE, 2018, 1–6
11. Liu Q, Chen X, Han B, et al. Virtual constraint based control of bounding gait of quadruped robots. *Journal of Bionics Engineering*, 2017, 14(2): 218–231
12. Khoramshahi M, Nasiri R, Shushtari M, et al. Adaptive natural oscillator to exploit natural dynamics for energy efficiency. *Robotics and Autonomous Systems*, 2017, 97: 51–60
13. Nasiri R, Khoramshahi M, Ahmadabadi M N. Design of a nonlinear adaptive natural oscillator: Towards natural dynamics exploitation in cyclic tasks. In: *Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon: IEEE, 2016, 3653–3658
14. Buchli J, Iida F, Ijspeert A J. Finding resonance: Adaptive frequency oscillators for dynamic legged locomotion. In: *Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Beijing: IEEE, 2006, 3903–3909
15. Buchli J, Ijspeert A J. Self-organized adaptive legged locomotion in a compliant quadruped robot. *Autonomous Robots*, 2008, 25(4): 331–347
16. Pratt G A, Williamson M M. Series elastic actuators. In: *Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Human Robot Interaction and Cooperative Robots. Pittsburgh: IEEE, 1995, 399–406
17. Pratt J E, Krupp B T. Series elastic actuators for legged robots. *Unmanned Ground Vehicle Technology VI*, 2004, 5422: 135–145
18. Junior A G L, de Andrade R M, Bento Filho A. Series elastic actuator: Design, analysis and comparison. In: Wang G, ed. *Recent Advances in Robotic Systems*. London: IntechOpen, 2016, 203–234
19. Spröwitz A, Tuleu A, Vespignani M, et al. Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. *International Journal of Robotics Research*, 2013, 32(8): 932–950
20. Azad M, Mistry M N. Balance control strategy for legged robots with compliant contacts. In: *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle: IEEE, 2015: 4391–4396
21. Canudas C, Roussel L, Goswami A. Periodic stabilization of a 1-DOF hopping robot on nonlinear compliant surface. *IFAC Proceedings Volumes*, 1997, 30(20): 385–390
22. Murphy K N, Raibert M H. Trotting and bounding in a planar two-legged mode. In: Morecki A, Bianchi G, Kdzior K, eds. *Theory and Practice of Robots and Manipulators*. Boston: Springer, 1985, 411–420
23. Raibert M H. Trotting, pacing and bounding by a quadruped robot. *Journal of Biomechanics*, 1990, 23: 79–98
24. Poulakakis I, Papadopoulos E, Buehler M. On the stability of the passive dynamics of quadrupedal running with a bounding gait. *International Journal of Robotics Research*, 2006, 25(7): 669–687
25. Zhou C, Wang B, Zhu Q, et al. An online gait generator for quadruped walking using motor primitives. *International Journal of Advanced Robotic Systems*, 2016, 13(6): 1729881416657960
26. Wang B, Wan Z, Zhou C, et al. A multi-module controller for walking quadruped robots. *Journal of Bionics Engineering*, 2019, 16(2): 253–263
27. Ijspeert A J, Nakanishi J, Hoffmann H, et al. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 2013, 25(2): 328–373
28. Ajallooeian M, van den Kieboom J, Mukovskiy A, et al. A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape. *Physica D: Nonlinear Phenomena*, 2013, 263: 41–56