J Bionic Eng 16 (2019) 621–632
DOI: https://doi.org/10.1007/s42235-019-0050-3

Journal of Bionic Engineering
http://www.springer.com/journal/42235

A Trot and Flying Trot Control Method for Quadruped Robot Based on Optimal Foot Force Distribution

Teng Chen, Xiaobo Sun, Ze Xu, Yibin Li, Xuewen Rong, Lelai Zhou*

Center for Robotics, *School of Control Science and Engineering*, *Shandong University*, *Jinan* 250061, *China*

Abstract

In order to enhance the dynamic motion capability of the bionic quadruped robot, a flying trot gait control method based on full-scale virtual model and optimal plantar force distribution is proposed. A stable flying trot gait is accomplished by mapping the robot torso motion to the foot trajectory. The force distribution calculated by the torso virtual model is converted into a quadratic optimization problem and solved in real time by the open source library Gurobi. The transition between the trot gait and the flying trot gait is achieved by coordinating leg motion phases. The results of the dynamic simulation verify that the proposed method can realize the 3D stable flying trot gait. Compared against the trot gait, the flying trot gait can improve the speed of the quadruped robot. Combine the trot gait and the flying trot gait, the quadruped robot can move efficiently and adapt to complex terrains.

Keywords: quadruped robot, flying trot, virtual model control, quadratic optimization

1 Introduction

The quadruped in the nature exhibits strong athletic ability and environmental adaptability. For example, the cheetah can run at a speed of more than 100 $km \cdot h^{-1}$, and the antelope can climb on a cliff close to 90 degrees. In order to achieve similar capabilities, motion control and environmental interaction methods for quadruped robots are important research contents to realize high dynamic characteristics.

The gait of the quadruped robots is mainly divided into static gait and dynamic gait. In the static gait, the robot has at least three legs as support legs at each moment, which means the legs can adjust the posture to ensure the stability of the robot in real time. The research on the static gait of the quadruped robots is systematic and mature. McGhee and Frank proposed the concept of static stability margin[1]. Messuri and Klein proposed the concept of Energy Stability Margin (ESM)[2]. Hirose standardized ESM and proposed the standard energy stability margin (NESM)[3]. Papadopoulos and Rey proposed the concept of force angle stability margin[4]. By applying these static stability control theories, the stable walking of the quadruped robots can be achieved[5]. Static gaits sacrifice the dynamic performance advantage of the robot to ensure the real-time stability of the system. The mobility under high-speed motion and complex terrain adaptation cannot be achieved.

In recent years, the high-speed motions of quadruped robots in the dynamic gait have become a research hotspot. The BigDog robot[7] developed by Boston Dynamics, the HyQ robot[8] of the Italian Institute of Technology, and the StarlETH robot[9] of ETH Zurich showed the fast motion capability under the trot gait. The MIT Cheetah robot developed by MIT uses the bound gait to achieve the speed of 6.4 $m \cdot s^{-1}$. Its Cost of Transpose (COT) is only 0.47, which indicates efficient high-speed motion. The WildCat robot developed by Boston Dynamics uses the galloping gait to achieve the current maximum speed of the quadruped robot at 8.9 $m \cdot s^{-1}$.

Quadruped mammals in the nature use asynchronous gaits at different motion speeds. For example, when the horse moves from low speed to high speed, the gait changes from walk to trot and then gallop. Studies have shown that each gait has the lowest energy consumption corresponding to the motion speed[10]. Quadrupeds use the trot or flying trot gait in the medium to high speed range, which corresponds to the minimum energy consumption. Based on these studies, this paper focuses on

***Corresponding author:** Lelai Zhou
E-mail: zhoulelai@sdu.edu.cn

the control method of the quadruped robot with trot and flying gaits.

The trot gait is a walking pattern in which one diagonal leg pair moves simultaneously, alternating with the other pair of legs. A flying trot is a special case characterized by a ballistic body motion, *i.e.* by a period in which there are no legs in contact with the ground. Because of many constraints such as structure, drive, control method, there are not many quadruped robots that have achieved flying trot. These robots can be divided into two categories according to whether there are elastic components on the leg structure. Raibert's quadruped robot from the CMU and later MIT leg lab was the first quadruped robot to demonstrate a flying trot. Around 20 years later, Raibert *et al*. presented BigDog[7], which has spring at the leg structure.

When the legs are inelastic, a flying trot gait can be achieved with the necessary to release the impact by using active compliance control method and force distribution[11,12]. The Reactive Control Framework (RCF), used by the HyQ robot, integrates the basic robot motion generator and controller to realize the robot reactive dynamic motion without environmental information. Based on the RCF method a stable flying trot is achieved by adjusting the parameters to match the resonant frequency determined by the mass and the stiffness of the leg in the robot's stance phase[8]. StarlETH shows the trotting with short flight phase[13]. Based on an online ZMP motion planner and continuously updates the reference motion trajectory as a function of the contact schedule, the ANYmal robot, developed by the ETH Zurich, can execute dynamic gaits including flying trot, jumping, pronking and others[14]. Recently, ANYmal has learned a new gait that can greatly improve the speed by deep reinforcement learning method, and this gait is between trot and flying trot gaits[15]. The Cheetah-cub[16] is a small quadruped robot with spring-loaded pantograph mechanism, which weights only 1.1 kg, and can demonstrate a flying trot. The MIT Cheetah robot[17,18] achieves a variety of gait including flying-trot by using Model Predictive Control (MPC).

In this work, a complete control approach for the quadruped robot to achieve stable flying trot gait is proposed. The proposed method is robust for the quadruped robot to achieve dynamic motion under disturbances. In detail, the advantages of the method can be summarized as follows:

(1) The proposed method can achieve a high-speed and stable flying trot gait of the quadruped robot. The framework consists of planning and control parts, which is modular and easy to implement for different platforms.

(2) The method proposed in this paper is robust, and the robot can resist a certain range of external forces under flying trot gait. Only a few control methods for the quadruped robot to achieve lateral recovery using the flying trot gait have been developed and presented publicly.

(3) The transition ability between the trot and the flying trot gait improve the robot's motion efficiency and terrain adaptability.

2 Materials and methods

2.1 Model of the quadruped robot

The model for study in this work is shown as Fig. 1a,

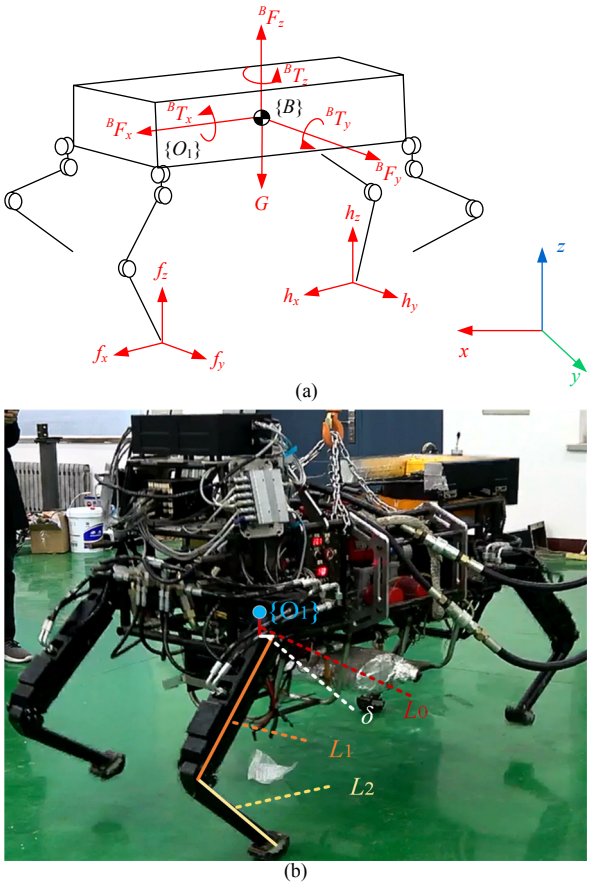

Fig. 1 Model of the quadruped robot. (a) Simplified model of the quadruped robot; (b) the SCalf-III robot prototype.

which is constructed based on the robot platform SCalf-III[19] built by Shandong University as shown in Fig. 1b. The Scalf-III is a hydraulic actuated quadruped robot, which is around 1.5 m long, 0.8 m wide and 1.3 m high with fully extended legs. The mass of the robot is about 300 kg. The onboard engine drives a variable pump and a hydraulic pressure station to provide stable 21 MPa pressure in the pipes. The body coordinate system $\{B\}$ is established at the center of the torso, and the leg coordinate system $\{O_i\}$ (i = 0, 1, 2, 3 represent the right front leg, left front leg, right hind leg and left hind leg, respectively) is established at the hip joint of each leg as shown in Fig. 1b. The SCalf-III has 4 identically constructed legs with the same parameters in the leg coordinate system. Each leg consists of 3 rotating joints that are the roll joint used to adduction and abduction at the hip (Hip A/A), the pitch joint used to flexion and extension at the hip (Hip F/E) and the pitch joint used to flexion and extension at the knee (Knee F/E). The linkages between the joints are donated by L_0, L_1, L_2, which are 45.0 mm, 450.9 mm and 404.8 mm long, respectively. Each joint is actuated by a hydraulic cylinder controlled by a servo valve with position and force sensors.

2.2 Flying trot gait planning

The quadruped robot motion planning is divided into the torso motion planning and the leg motion mapping. The target motion of the robot is obtained by fitting and optimizing the position and velocity at the key points and connecting them by an interpolation algorithm.

2.2.1 Torso motion planning

The movement of the torso during each motion cycle consists of two repetitive motion trajectories, as shown in Fig. 2. The Left Front (LF) leg and the Right Hind (RH) leg on the diagonal line are grouped as G1. The Right Front (RF) leg and the Left Hind (LH) leg are grouped as G2. The two groups alternately perform the stance phase and the swing phase motions. When G1 is in the stance phase, the legs are buffered and then bounced to achieve compliance touch and then pops up. The torso motion in the sagittal plane has the trend as (0, t_1) in Fig. 2. The legs reach the target lift height and then swing to the target landing position. When G1 is in the swing state, the robot will drop due to gravity, then

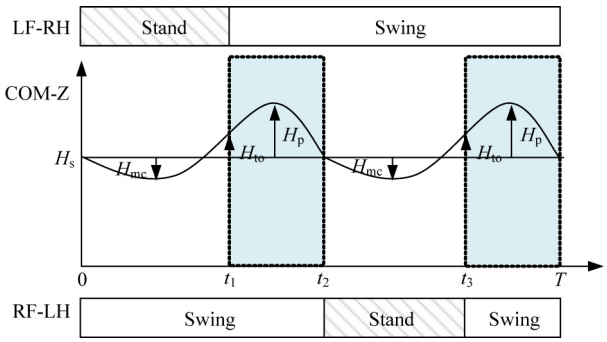

Fig. 2 COM trajectory of the quadruped with flying trot gait.

G2, which is originally in the swing state, enters the stance state after touching the ground at the time t_2. Then G2 repeats the motion process performed by G1 above to complete the motion of the robot for one cycle.

$$\begin{cases} t_{\text{sky}} = (-\sqrt{2g(H_{\text{to}} - H_{\text{p}})} - \sqrt{-2gH_{\text{p}}}) / g \\ t_{\text{stand}} = (1/f - 2t_{\text{sky}}) / 2 \\ t_{\text{swing}} = 1/2f + t_{\text{sky}} \\ v_{\text{to}}^{\text{B}} = -\sqrt{2g(H_{\text{to}} - H_{\text{p}})} \\ v_{\text{td}}^{\text{B}} = v_{\text{to}}^{\text{B}} + gt_{\text{sky}} \end{cases}, \quad (1)$$

where t_{stand} is the stance phase time, t_{sky} is the vacant time, t_{swing} is the swing time, v_{to}^{B} is the torso velocity when the robot is off the ground and v_{td}^{B} touch the ground, f is the step frequency of the robot, g is the gravitational acceleration and in this work it is −9.8 m·s^{-2}.

2.2.2 Leg motion planning

The foot motion planning method for the x and y directions of the robot leg is the same. The leg motion is planned in leg coordinate system. Based on Raibert's planning method[20], the velocity and position of the swing leg along x direction v_x^{L} and p_x^{L} are planned as:

$$\begin{cases} v_x^{\text{L}} = -\dot{x}_{\text{d}} \\ p_x^{\text{L}} = \dfrac{\dot{x}_{\text{d}} T_{\text{f}}}{2} + k_{xv}(\dot{x} - \dot{x}_{\text{d}}) \end{cases}, \quad (2)$$

where $\dot{x}_{\text{d}}$ and $\dot{x}$ represent the target velocity and current velocity of the robot in the x direction, respectively. T_{f} represents the swing time, and k_{xv} is the velocity compensation coefficient. Derivation and detailed description of the equation are in Raibert's book.

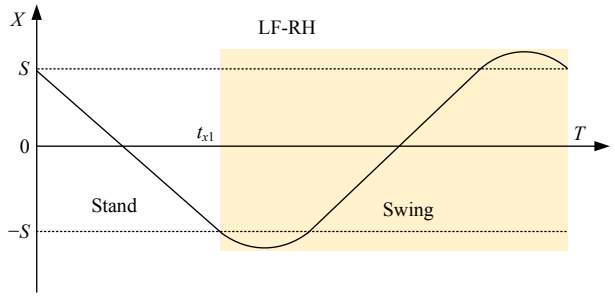

Fig. 3 Leg trajectory along x axis of the quadruped robot in leg coordinate system.

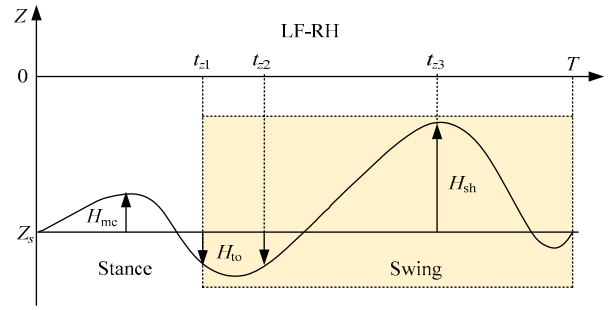

Fig. 4 Leg trajectory along z axis of the quadruped robot.

Table 1 Flying trot motion parameter design

Parameters	Symbols	Values (mm)
Stand height of the body in $\{W\}$ frame	H_s	810
Stand height of the leg in $\{O_i\}$ frame	L_s	−710
Peak point height	H_p	$\geq H_{to}$
Takeoff point height	H_{to}	≥ 0
Swing height	H_{sh}	≥ 0
Compression height of the leg	H_{mc}	–

Fig. 3 shows the trajectory along x axis, where S is the step length calculated by the desired robot velocity. In this work, the movement of the stance phase is achieved by the virtual force detailed in the following sections, so no trajectory planning is performed. The trajectory in the stance phase is drawn together for the complete display in Fig. 3.

The leg movement process of the quadruped robot in the vertical direction can be divided into four parts as shown in Fig. 4, where Z_s is the stand height of the leg in O_i frame, which equals to L_s in Table 1. First, when the leg touches the ground, the leg is compressed until reaches the maximum compression H_{mc} and then begins to elongate. The Center of Mass (COM) of the robot begins to rise. At time t_{z1} (equal to the time t_1 in Fig. 2), the leg leaves the ground and the leg state changes to the swing state. The torso velocity at the time of taking off the ground v_{to}^{B} is planned by the torso movement detailed in section 2.2.1. This velocity is the reverse velocity provided by the leg movement, so the leg velocity at the time of leaving the ground t_{z1} is $-v_{to}^{B}$. Second, the leg velocity at the time of leaving the ground is the same to the torso velocity, however, the leg still has large elongation velocity because of movement inertia when starts the swing state. In order to accelerate the withdrawal process to lift the leg quickly, the leg velocity at the time t_{z2} is designed to be λ_2 (λ_2 is minus) times the torso velocity. The withdrawal acceleration is λ_3 (λ_3 is minus) times the gravity acceleration g. Third, the leg raises to the desired height during time t_{z2} to t_{z3}. When the leg reaches the highest position at the time of t_{z3}, there is no speed at the z-direction. Fourth, from t_{z3} to the end of the swing period is the leg-down process, during which time the leg recovers the target standing height. Then the leg waits to land and becomes the stance leg. The time for lifting and dropping the leg is divided by a parameter λ_1. When the leg changes from the swing state to the stance state, the leg velocity is designed to be λ_0 (λ_0 is minus) times the corresponding torso velocity v_{td}^{B} to achieve active buffering landing. Then the leg motion enters the next cycle. According to the above motion planning, the position and velocity of the leg movement at 5 key-points are obtained:

$$\begin{cases} v_{z0}^{L} = v_{td}^{B}\lambda_0 \\ v_{z1}^{L} = -v_{to}^{B} \\ v_{z2}^{L} = v_{z1}^{L}\lambda_2 \\ v_{z3}^{L} = 0 \\ v_{z4}^{L} = v_{z0}^{L} = v_{td}^{B}\lambda_0 \end{cases}, \tag{3}$$

$$\begin{cases} p_{z0}^{L} = z_s \\ p_{z1}^{L} = p_{z0}^{L} - H_{to} \\ p_{z2}^{L} = p_{z1}^{L} + (v_{z2}^{L}\cdot v_{z2}^{L} - v_{z1}^{L}\cdot v_{z1}^{L})/(2g\lambda_3), \\ p_{z3}^{L} = p_{z0}^{L} + H_{sh} \\ p_{z4}^{L} = p_{z0}^{L} \end{cases} \tag{4}$$

$$\begin{cases} t_{z1}^{L} = (v_{z2}^{L} - v_{z1}^{L})/(g\lambda_3) \\ t_{z2}^{L} = (1/f - t_{stand} - v_{z1}^{L})\lambda_2 \\ t_{z2}^{L} = (1/f - t_{stand} - v_{z1}^{L})\cdot(1-\lambda_2) \end{cases}. \tag{5}$$

2.2.3 Trajectory generation

The trajectory can be regarded as the composition of multiple small trajectory segments conducted by position and velocity constraints discussed above. After knowing the position (y_{init}, y_{end}) and velocity ($\dot{y}_{\text{init}}$, $\dot{y}_{\text{end}}$) of the initial and final points of each track, the cubic interpolation model function is used to achieve smooth connection and transition of each segment:

$$\begin{cases} y = k_{\text{a}}t^3 + k_{\text{b}}t^2 + k_{\text{c}}t + k_{\text{d}} \\ k_{\text{d}} = y_{\text{init}} \\ k_{\text{c}} = \dot{y}_{\text{init}} \\ k_{\text{a}} = (\dot{y}_{\text{end}}t - 2y_{\text{end}} + \dot{y}_{\text{init}}t + 2y_{\text{init}}) / T^3 \\ k_{\text{b}} = (\dot{y}_{\text{end}} - 3k_{\text{a}}T^2 - k_{\text{c}}) / (2T) \end{cases}. \tag{6}$$

In the trajectory planning section, the target position and corresponding velocity can be obtained. The starting point position and velocity can be obtained through the feedback of the robot. The designed trajectory can be smoothly connected by an interpolation algorithm.

2.3 Flying trot controller design for the quadruped robot

The virtual model controller and its corresponding optimal foot force distribution method are constructed based on the full-scale model of the quadruped robot in the stance state. The leg virtual model control method is used for trajectory tracking in the swing state.

2.3.1 Virtual model controller in the stance state

Since the mass of the robot leg is small compared to the torso (less than 1:15), the leg is modelled massless in the whole-body virtual model. The robot torso geometric center *B* as shown in Fig. 1a is used as the origin of the lumped mass model to simplify the modelling process. For quadruped robots with trot or flying trot gait movement, the diagonal legs alternately perform a stance phase and a swing phase motion. The diagonal legs in the stance phase can be G1 or G2 as defined in section **2.1**, and there is always one front leg and one hind leg to provide support points and corresponding supporting forces. During the steady motion of the robot, the static force analysis of the legs in the stance state is carried out. The torso is supported by the forces and torques generated from the front and hind supporting legs. The virtual centroid force matrix regarded as $[{}^{\text{B}}\boldsymbol{F}\ {}^{\text{B}}\boldsymbol{T}]^{\text{T}} = [{}^{\text{B}}F_x\ {}^{\text{B}}F_y\ {}^{\text{B}}F_z\ {}^{\text{B}}T_x\ {}^{\text{B}}T_y\ {}^{\text{B}}T_z]^{\text{T}}$ is distributed to the supporting legs to obtain the following representation:

$$\begin{bmatrix} {}^{\text{B}}\boldsymbol{F} \\ {}^{\text{B}}\boldsymbol{T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{p}_{\text{f}} \times & \boldsymbol{p}_{\text{h}} \times \end{bmatrix} \begin{pmatrix} \boldsymbol{F}_{\text{f}} \\ \boldsymbol{F}_{\text{h}} \end{pmatrix}, \tag{7}$$

where $\boldsymbol{F}_{\text{f}} = [f_x\ f_y\ f_z]^{\text{T}}$ stands for the virtual force that need to be applied to the front stance leg, and $\boldsymbol{F}_{\text{h}} = [h_x\ h_y\ h_z]^{\text{T}}$ is for hind stance leg. $\boldsymbol{p}_{\text{f}} = [x_F\ y_F\ z_F]^{\text{T}}$ is the position vector of the support point of the front leg in stance phase relative to the centroid point, and $\boldsymbol{p}_{\text{h}} = [x_H\ y_H\ z_H]^{\text{T}}$ is for the hind supporting point. $\boldsymbol{I}$ is an unit matrix.

According to the robot structure and leg position, the analysis format of the virtual force relationship between the torso and the supporting leg can be summarized as:

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}, \tag{8}$$

which can be expressed as follows:

$$\begin{bmatrix} {}^{\text{B}}F_x - G_{\text{x}} \\ {}^{\text{B}}F_y - G_y \\ {}^{\text{B}}F_z + G_z \\ {}^{\text{B}}T_x \\ {}^{\text{B}}T_y \\ {}^{\text{B}}T_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & -z_{\text{F}} & y_{\text{F}} & 0 & -z_{\text{H}} & y_{\text{H}} \\ z_{\text{F}} & 0 & -x_{\text{F}} & z_{\text{H}} & 0 & -x_{\text{H}} \\ -y_{\text{F}} & x_{\text{F}} & 0 & -y_{\text{H}} & x_{\text{H}} & 0 \end{bmatrix} \begin{bmatrix} f_x \\ f_y \\ f_z \\ h_x \\ h_y \\ h_z \end{bmatrix}, \tag{9}$$

where $G_x = m_{\text{B}}g\sin(\beta), G_y = 0, G_z = m_{\text{B}}g\cos(\beta)$.

The roll angle of the torso is α and the pitch angle is β. m_{B} stands for robot mass. The decomposition method is based on the assumption that the roll angle remains horizontal all the time. In order to achieve a stable torso movement, the torso virtual centroid force is designed based on the whole-body virtual spring-damping model[21]. The force and torque at the center of mass of the torso are designed as a spring-damp model as:

$$\begin{cases} {}^{\text{B}}F_x = k_x(\dot{x}_{\text{d}} - \dot{x}) + k_{\text{ffx}}\dot{x}_{\text{d}} \\ {}^{\text{B}}F_y = k_y(\dot{y}_{\text{d}} - \dot{y}) + k_{\text{ffy}}\dot{y}_{\text{d}} \\ {}^{\text{B}}F_z = k_z(h_{\text{d}} - h) + \dot{k}_z(\dot{h}_{\text{d}} - \dot{h}) \\ {}^{\text{B}}T_x = k_\alpha(\alpha_{\text{d}} - \alpha) + \dot{k}_\alpha(\dot{\alpha}_{\text{d}} - \dot{\alpha}) \\ {}^{\text{B}}T_y = k_\beta(\beta_{\text{d}} - \beta) + \dot{k}_\beta(\dot{\beta}_{\text{d}} - \dot{\beta}) \\ {}^{\text{B}}T_z = k_\omega(\omega_{\text{d}} - \omega) \end{cases}, \tag{10}$$

where $\dot{x}_d$ and $\dot{y}_d$ represent the target velocity along x and y axis, respectively. h_d and $\dot{h}_d$ are the target centroid height and centroid movement velocity, which are related to the P_{zi}^L and v_{zi}^L introduced in the section 2.2.2, respectively. α_d, β_d and ω_d are the target roll angle, pitch angel and yaw velocity, separately. The same label without subscript d indicates the actual variable of the robot. k_x, k_y and k_z are the stiffness coefficients for the velocity compensation along x, y, z axes. k_{ffx} and k_{ffy} are the velocity feedforward coefficients in x and y directions, respectively. k_α, k_β and k_ω are the stiffness coefficients for the torso angle compensation along rotation in x, y, z axes, respectively. $\dot{k}_\alpha$ and $\dot{k}_\beta$ are the damping coefficients for the torso velocity compensation along rotation in x, y axes, respectively.

2.3.2 Optimal force distribution in stance state

In Eq. (8), $\boldsymbol{A}$ is a singular matrix. There is no definite solution for the force distribution of the centroid virtual force to the support legs. Many researchers propose different solutions to solve this problem. Zhang *et al.*[21] discards the lateral force, which means applying the constraint $f_y = h_y$, to obtain the distribution of the foot force. This method is consistent with the phenomenon that animals rarely use lateral motion in nature. However, this method cannot achieve full-dimensional force control of the robot. E *et al*. use weighted average vertical direction force to achieve force distribution, but this method can't achieve reasonable force distribution under unstructured terrains[22]. In this work, the force distribution is transformed into a quadratic programming problem. In order to make the foot force distribution meet the requirements of the centroid virtual force and torque, while minimizing the output force to achieve efficient energy utilization, the quadratic optimization problem is constructed as:

$$\begin{aligned} &\min\ Q = (\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^{\mathrm{T}}\boldsymbol{S}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{x}, \\ &\text{s.t. } |f_x| < \mu f_z, |f_y| < \mu f_z, |h_x| < \mu h_z, \\ &\quad |h_y| < \mu h_z, f_z \geq 0, h_z \geq 0, \end{aligned} \tag{11}$$

where $\boldsymbol{S}$ = diag(6) is a weight diagonal matrix used to make the output follow the target virtual force. $\boldsymbol{W}$ = diag(6) denotes the penalty diagonal matrix, so as to avoid the occurrence of the excessive virtual force. μ represents the friction coefficient. In the constraint term, the support force is guaranteed to be positive, and the friction cone constraint is applied to ensure that the tangential force is within the static friction range.

Table 2 Control parameters of the robot motion

Parameters	Value	Unit
T_s	0.34	s
H	0.15	m
k_x	500	$\mathrm{N\cdot(m\cdot s^{-1})^{-1}}$
k_z	80000.0	$\mathrm{N\cdot m^{-1}}$
k_α	20000.0	$\mathrm{N\cdot m^{-1}}$
k_{px}	10000.0	$\mathrm{N\cdot m^{-1}}$
k_{pz}	20000.0	$\mathrm{N\cdot m^{-1}}$
$\boldsymbol{S}$	diag{10, 10, 200, 10, 10, 10}	
$\boldsymbol{W}$	diag{0.1, 0.1, 0.1, 0.1, 0.1, 0.1}	
λ_0	−0.1	–
λ_1	0.6	–

The optimization variable is the 6-dimensional $\boldsymbol{x}$ as defined in Eq. (8). In this work, the quadratic optimization problem is solved by the open source library Gurobi. By solving the problem of virtual force distribution in the full-scale model, the number of optimization functions is simplified, and the solution is accelerated. This guarantees real-time online optimization of the force distribution.

The swing leg controller is based on the virtual model of the leg[23], and the swing leg is equivalent to the spring-damping model. The 3-dimensional virtual force of the swing leg is calculated as the spring-damp model as:

$$\begin{cases} f_x = k_{px}(x_d - x) + k_{dx}(\dot{x}_d - \dot{x}) \\ f_y = k_{py}(y_d - y) + k_{dy}(\dot{y}_d - \dot{y}), \\ f_z = k_{pz}(z_d - z) + k_{dz}(\dot{z}_d - \dot{z}) \end{cases} \tag{12}$$

where k_{px} and k_{dx} are stiffness and damping of the swing leg along x axis, respectively. Other dimension parameters follow the same naming way. The end effector force is converted to the joint space servo torque by the leg Jacobian matrix.

$$\boldsymbol{\tau}_{vm} = \boldsymbol{J}^{\mathrm{T}}\boldsymbol{f}, \tag{13}$$

where $\boldsymbol{f} = [f_x\, f_y\, f_z]^{\mathrm{T}}$ is the force vector of the swing leg. Detailed information about the virtual model control of the leg and the application for the SCalf serial robot can refer[24].

The basic control framework is shown in Fig. 5.

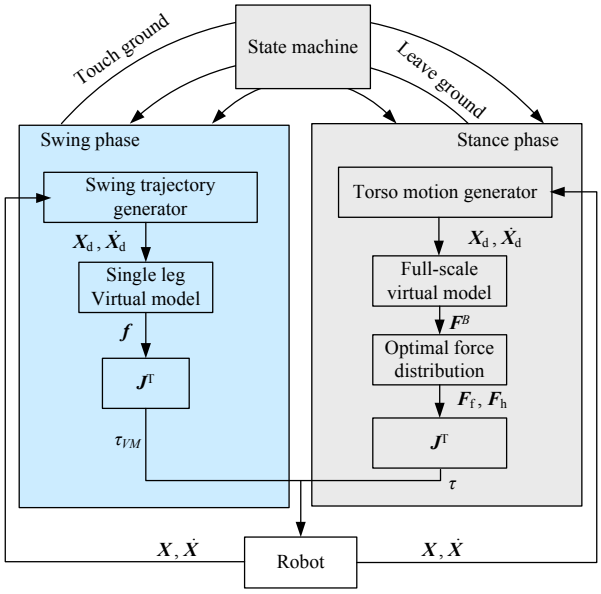

Fig. 5 Overview of the control architecture. $\boldsymbol{X}$ is the vector consist of the position in the x, y and z axis direction

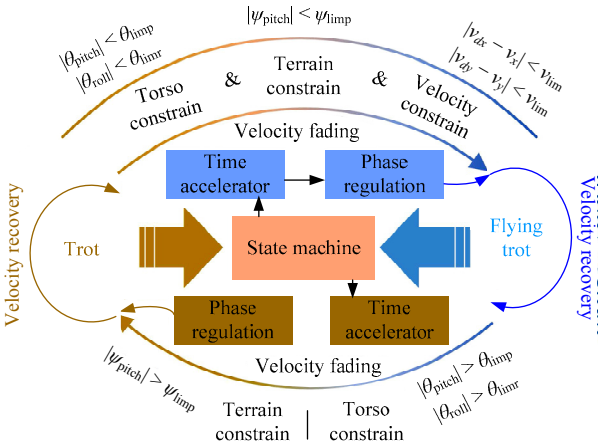

Fig. 6 Transition between trot and flying trot gaits.

The overall motion of the robot is divided into stance and swing phases, which alternately form a diagonal gait. The state machine coordinates the state transition among the legs according to the planning and the touchdown information of each foot.

2.3.3 States transition

The stance and swing states of the diagonal legs of the robot are determined by the combination of the time state machine[20] and the ground contact information. Time state machine is the leg state transition module, which coordinate the leg state transition between stance state and swing state based on timer. The stance time and swing time of the diagonal legs can be calculated according to the robot step frequency and the flying vacant information detailed in the section **2.2**. In the case of flat terrain without disturbance, the diagonal leg state of the robot switches according to this time state machine. When the terrain has an undulating change or the robot is disturbed by external forces, the touchdown information is merged to perform state transition. When the leg in the swing state touches the ground before the time state machine transition, the state of the leg is switched to the support phase. The stiffness and damping parameters of the leg in the swing phase are linearly increased to the support phase target parameter to achieve the support motion. When the support phase leaves the ground early without reaching the swing state time, the stance leg advances into the swing state, and the virtual stiffness of the leg linearly changes to a smaller swing phase stiffness to prepare the swing motion.

2.3.4 Gait transition between trot and flying trot

The ability of transition between the trot and flying trot gaits for quadruped robots improves the environment adaptiveness and motion efficiency. Flying trot can let the robot run faster, but have limited robustness to complex terrains and disturbances when compared to the trot gait. The combination of the trot and flying trot gaits can make the quadruped robot walk agilely and consume low energy.

The trot gait control method is the same as the flying trot except that H_p and H_{to} are set to zero. The method of the gait transition strategy in this paper is shown in Fig. 6. The speed of the robot is reduced before the gait transition to weaken the disturbance caused by the inertia change. The transection from flying trot to trot need reduce the motion speed first because the maximum speed in the trot gait is less than the flying trot. Moreover, due to the rapid adjustment of the motion phase of the robot's leg during the gait change, the inertia of the robot changes greatly, and maintaining the gait transition at a high speed will make the robot unstable. The transition from trot to flying trot follow the same mechanism mainly because the inertia of the transformation will have an effect on the steady state of the robot. The state time variables of the new gait are calculated before the transition, and the time accelerator

coordinates the leg acceleration to the minimum leg fluctuation time to complete the transition. The robot can automatically or manually switch to flying trot after a period of steady trot motion on non-rough terrain. The conditions for stable switching are as follows: (a) The torso pitch and roll angle of the robot are kept within the small fluctuations θ_{limp} and θ_{limr}. (b) The terrain inclination is suitable for the flying trot, which is judged by the laser placed downward 30° in front of the robot. (c) The robot is currently moving stably, which means the fluctuations of the velocities along x and y axis are less than v_{lim}.

When the robot encounters a rugged environment or an external force disturbance during the flying trot movement, the robot can automatically switch to the trot gait. The switching condition can be any one of the following conditions. (a) The terrain inclination is larger than ψ_{limp}. (b) The robot torso attitude angle exceeds the critical stability angle θ_{limp} or θ_{limr}.

2.3.5 Terrain adaptation

To improve the stability of the quadruped robot on slopes, a sample adaptation method[24] is used in this work. The torso pitch angle is calculated by the legs in the stance state based on the gradient of the virtual slope[21]. The swing trajectory is offset by a coefficient with the pitch angle measured by the Inertial Measurement Unit (IMU). These two modulations make the robot climb slopes with the maximal inclination of 30°.

3 Results

The robot model with the same size and inertia as the Scalf-III physical platform is established in the Webots PRO R2018a simulation software. The position and force outputs of each joint are constrained to be the same as the actual platform. The feasibility of the proposed method is discussed through simulations.

3.1 Omnidirectional flying trot

All movements of the robot are the composition of the basic forward-backward, left-right, and steering movements. In order to test the performance of the proposed control method on the omnidirectional motion of the robot, the robot target motion is the combination of the 0.5 $\text{m}\cdot\text{s}^{-1}$ forward velocity, the 0.1 $\text{m}\cdot\text{s}^{-1}$ lateral velocity and the 0.1 $\text{rad}\cdot\text{s}^{-1}$ anticlockwise steering motion. The robot moves along a circle track from the original position according to the above target. After returning to the original point, the lateral velocity and the steering angular velocity become negative, namely −0.1 $\text{m}\cdot\text{s}^{-1}$ and −0.1 $\text{rad}\cdot\text{s}^{-1}$. The trajectory obtained by the simulation with respect to the desired trajectory is represented in Fig. 7. The robot starts the flying trot motion after receiving the motion command as shown in Fig. 8. The robot accelerates from the stationary state to the target motion velocity. Due to the inertia and friction problem, the robot will have a small deviation from the expected motion trajectory at this stage. When the robot returns back after tracking a circle, the target moving velocity is adjusted so that the robot reverses the steering motion. Since the trotting velocity can not completely follow the target velocity during the motion, the velocity reverse is constrained both by the timer and the body position. The switching process does not add the transition part, so there will be an obvious disturbance at the velocity switching point as shown in Fig. 8.

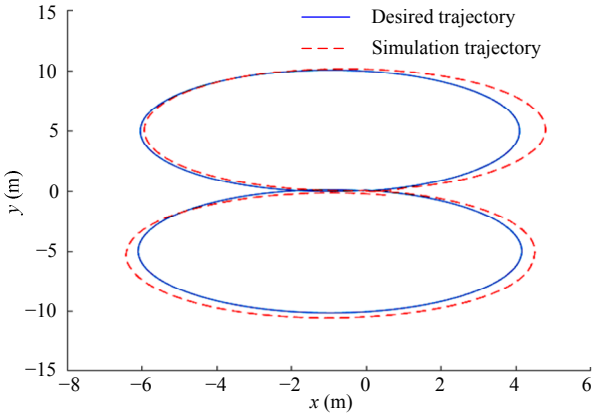

Fig. 7 Omnidirectional moving trajectory following curve.

After completing the whole process, the robot finishes an "8" shape. When the robot returns to the x-axis of the initial position, the y-axis error is −0.239 m. The causes of the error are summarized as follows. First, the measurement error of the robot motion velocity is the key factor. An inertia filter is used in the control algorithm to smooth the velocity calculated by the leg position feedback. Secondly, the robot velocity will not update when there is no leg in contact with the ground, which causes the velocity estimation error. Finally, there will be sliding between the foot and the ground during

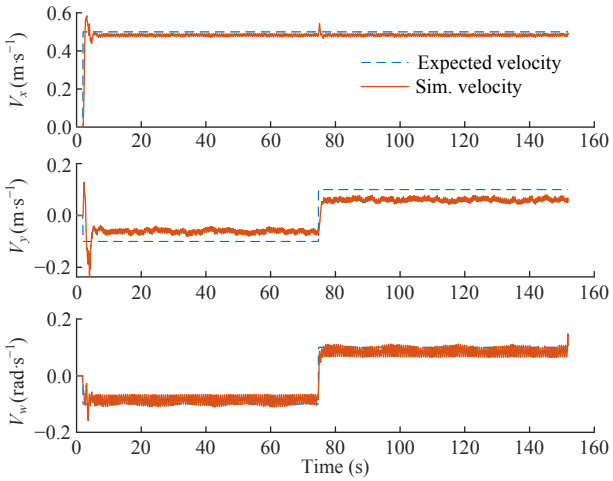

Fig. 8 Omnidirectional moving speed following curve.

the motion, and the inertia of the robot changes greatly in different states. There are many ways to avoid slipping, which is beyond the scope of this paper.

As shown in Fig. 8, there is velocity tracking static error, especially in the *y*-direction. This proves from the side that the quadrupeds in nature rarely use lateral motions, but uses the sagittal motion and turning to achieve equivalent lateral motions.

3.2 Lateral impact recovery under flying trot

The motion controller presented in this paper has an adaptive adjustment ability to external force disturbances. When disturbances cause the variations of the robot torso posture, the virtual force distribution of the whole-body virtual model is adjusted according to the Eq. (10). On the other hand, disturbances cause the unexpected velocity of the robot, which converges through closed-loop modification of the landing position in the swing state detailed in section **2.2**. Based on the above treatment the stable motion of the robot is guaranteed when encountering disturbance.

According to the structural characteristics of the quadruped robot, the lateral stability of the robot is worse than the sagittal stability, which means that the lateral disturbance of the robot is more likely to make the robot unstable. To test the robustness of the control system, a 60 kg weight pendulum is placed 1.4 m above the ground, where the robot torso is 0.85 m. The pendulum can rotate freely to bump the robot. The ability of the robot to resist external force disturbance is constrained in many ways. Firstly, the joint torque is limited.

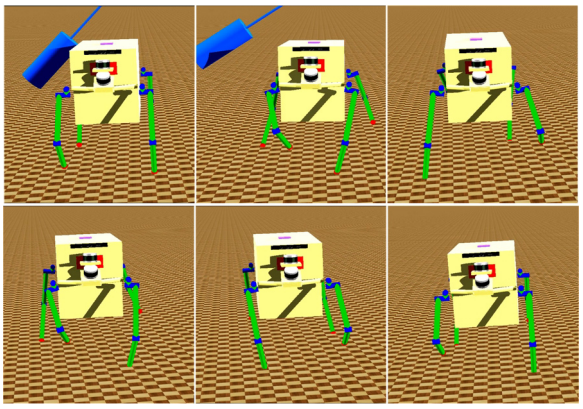

Fig. 9 Lateral impact adjustment snapshots.

Table 3 Lateral impact recovery method comparison

Robot	Gait	Support points
BigDog	Trot	2
AnyMal	Trot/walk	2 or 3
HyQ	Trot/walk	2 or 3
SCalf-II	Trot	2
DOGO II[25]	Trot	2
Ours	Trot/flying trot	2 or 0

The greater the disturbance, the greater the instantaneous joint torque is needed. The limited driving moment will restrict the adjusting ability. Secondly, the limited range of joint angle constrains the amplitude of the adjustment. In this simulation, the pendulum over 60 kg will cause the robot to topple over.

The adjustment process when encountering disturbance is shown in Fig. 9. The torso attitude angle and adjustment speed during the adjustment process are shown in Fig. 10. The robot is tilted by a lateral angle about 10° at 3 s, and the robot returns to stability after about 4 s. It can be seen that the proposed control method has better robustness against external disturbances. We summarized some typical quadruped robots that can achieve lateral disturbance recovery, and compared their implementation gaits as shown in Table 3. Methods for the quadruped robot to achieve lateral recovery using the flying trot gait are seldom, and our work contributes it.

3.3 Gaits transition and slope climbing

As detailed in section 2.3.4, the gaits transition is shown in Fig. 11. The constraint parameters of the robot for gait switching are $\theta_{\text{limp}} = 0.034$ radian, $\theta_{\text{limr}} = 0.026$ radian, $\psi_{\text{limp}} = 0.15$ radian, $v_{\text{lim}} = 0.1$ m·s^{-1}. The robot

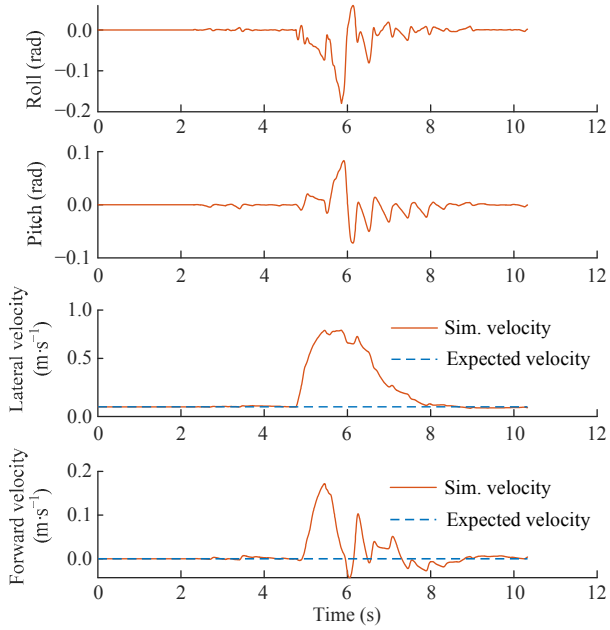

Fig. 10 Lateral impact velocity adjustment process.

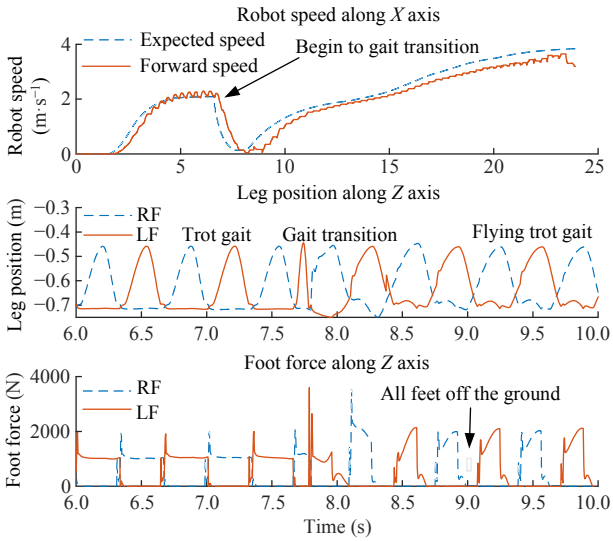

Fig. 11 Transition process between flying trot to trot gaits.

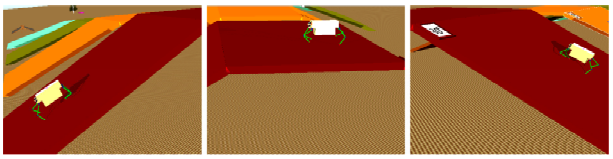

Fig. 12 30° slope up and down process of the robot.

began to change the gait from trot to flying trot as show in the upper subfigure of Fig. 11. The velocity decreases and the time accelerator modulate the motion phase to realize the change. The foot force in z-axis as shown in the middle of Fig. 11 proves the stable flying trot motion. There is vacant time at the intersection of the support state of the legs, which means zero contact force with all legs at the same time. The robot can run at a stable 3.2 $m \cdot s^{-1}$ by the flying trot gait while 2 $m \cdot s^{-1}$ by the trot gait with 1.5 Hz in this work. The trot gait is more robust when encountering slopes, stairs or external forces. The transition ability makes the robot be able to go through a large range of terrains efficiently.

As shown in Fig. 12 the robot can go through 30° slope up and down using trot gait with the terrain adaptation detailed in section 2.3.5. Without the adaptation, 20° slope is even a great challenge for the robot to climb. More inclination slopes are failed because of the lack of friction between the feet and the slopes.

During the prototype experiment, it is found that when the center of gravity of the robot is out of center, the robot will be twisted left and right to make the robot unstable. In the robot design, the mass should be distributed as evenly as possible, and the small-scale centroid offset can be corrected by attitude. This is also the direction of our future efforts. This method needs many parameters of the prototype platform, such as the mass distribution, body inertia and load variation. There are still difficulties to measure these parameters in the prototype. The demonstration of the method on the prototype will be shown in the future work.

4 Conclusion

In this paper, a flying trot gait control method is proposed for the bionic quadruped robot. The torso motion is planned based on the bionic movement and energy conservation. The planned motion is mapped to the legs and is achieved by the actuators on the legs. The virtual force calculated by the whole-body virtual model is distributed to the feet by optimizing the quadratic optimization problem, and then converted into joint torques. The method has certain adaptability to the external disturbance, and can realize the robust flying trot under external forces. The proposed method can realize the transition of the flying trot and trot gait to improve the movement efficiency at the complex terrains. The main contribution of this paper is a robust trot and flying trot gait control algorithm for quadruped robots. The method can realize the high-speed stable motion and the

adaptation to complex terrains for the quadruped robot.

Acknowledgment

This work was supported by the National Key R&D Program of China (Grant No. 2017YFC0806505), the National High-tech R&D Program of China (Grant No. 2015AA042201), National Natural Science Foundation of China (Grant Nos. U1613223 and 61603216) and the Key R&D Program of Shandong (Grant No. 2017CXGC0901).

References

[1] McGhee R B, Frank A A. On the stability properties of quadruped creeping gaits. *Mathematical Biosciences*, 1968, **3**, 331–351.

[2] Messuri D, Klein C. Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion. *IEEE Journal on Robotics and Automation*, 1985, **1**, 132–141.

[3] Hirose S, Tsukagoshi H, Yoneda K. Normalized energy stability margin and its contour of walking vehicles on rough terrain. *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*), Seoul, Korea, 2001, 181–186.

[4] Papadopoulos E, Rey D A. A new measure of tipover stability margin for mobile manipulators. *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*), Minneapolis, USA, 1996, 3111–3116.

[5] Zhang S S, Rong X W, Li Y B. A free gait generation method for quadruped robots over rough terrains containing forbidden areas. *Journal of Mechanical Science and Technology*, 2015, **29**, 3983–3993.

[6] Pongas D, Mistry M, Schaal S. A robust quadruped walking gait for traversing rough terrain. *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*), Roma, Italy, 2007, 1474–1479.

[7] Raibert M, Blankespoor K, Nelson G, Playter R. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 2008, **41**, 10822–10825.

[8] Semini C, Barasuol V, Boaventura T, Frigerio M, Focchi M, Caldwell D G, Buchli J. Towards versatile legged robots through active impedance control. *The International Journal of Robotics Research*, 2015, **34**, 1003–1020.

[9] Hutter M, Sommer H, Gehring C. Quadrupedal locomotion using hierarchical operational space control. *International Journal of Robotics Research*, 2014, **33**, 1047–1062.

[10] Hoyt D F, Taylor C R. Gait and the energetics of locomotion in horses. *Nature*, 1981, **292**, 239–240.

[11] Semini C, Barasuol V, Boaventura T, Frigerio M, Buchli J. Is active impedance the key to a breakthrough for legged robots? *Robotics Research*, 2016, **114**, 3–19.

[12] Palmer L R, Orin D E. Force redistribution in a quadruped running trot. *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*), Roma, Italy, 2007, 4343–4348.

[13] Gehring C, Coros S, Hutter M, Bellicoso C D, Heijnen H, Diethelm R, Bloesch M, Fankhauser P, Hwangbo J, Hoepflinger M A, Siegwart R. Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics and Automation Magazine*, 2016, **23**, 34–43.

[14] Bellicoso C D, Jenelten F, Gehring C, Hutter M. Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. *IEEE Robotics and Automation Letters*, 2018, **3**, 2261–2268.

[15] Hwangbo J, Lee J, Dosovitskiy A, Bellicoso D, Tsounis V, Koltun V, Hutter M. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 2019, **4**, eaau5872.

[16] Spröwitz A, Tuleu A, Vespignani A, Ajallooeian M, Badri E, Ijspeert J. Towards dynamic trot gait locomotion: Design, control, and experiments with Cheetah-cub, a compliant quadruped robot. *International Journal of Robotics Research*, 2013, **32**, 932–950.

[17] Hyun D J, Seok S, Lee J, Kim S. High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *The International Journal of Robotics Research*, 2014, **33**, 1417–1445.

[18] Di Carlo J, Wensing P M, Katz B, Bledt G, Kim S. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems* (*IROS*), Madrid, Spain, 2018, 1–9.

[19] Yang K, Zhou L L, Rong X W, Li Y B. Onboard hydraulic system controller design for quadruped robot driven by gasoline engine. *Mechatronics*, 2018, **52**, 36–48.

[20] Raibert M H, Tello E R. Legged robots that balance. MIT Press, Boston, USA, 1986, 57–80.

[21] Zhang G T, Rong X W, Hui C, Li Y B, Li B. Torso motion control and toe trajectory generation of a trotting quadruped robot based on virtual model control. *Advanced Robotics*, 2016, **30**, 284–297.

[22] E M C, Liu H, Zhang X L, Fu L, Ma H X. Compliant gait generation for a quadruped bionic robot walking on rough terrains. *Robot*, 2014, **36**, 584–591. (in Chinese)

[23] Pratt J, Chew C M, Torres A, Dilworth P, Pratt G. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research*, 2001, **20**, 129–143.

[24] Chen T, Rong X W, Li Y B, Ding C, Chai H, Zhou L L. A compliant control method for robust trot motion of hydraulic actuated quadruped robot. *International Journal of Advanced Robotic Systems*, 2018, **15**, 1729881418813235.

[25] Gonzalez-Luchena I, Gonzalez-Rodriguez A G, Gonzalez-Rodriguez A, Adame-Sanchez C, Castillp-Garcia F J. A new algorithm to maintain lateral stabilization during the running gait of a quadruped robot. *Robotics and Autonomous Systems*, 2016, **83**, 57–72.

Appendix

Derivative of Eq. (1)

From Fig. 2 the t_{sky} is consisted of two part. The rising time is from t_1 to the peak point, which can be calculated by:

$$H_p - H_{to} = \frac{1}{2}(-g)t_{\Delta 1}^2. \tag{A1}$$

And the time from the peak point to the t_2 can be calculated by:

$$H_p = \frac{1}{2}(-g)t_{\Delta 2}^2. \tag{A2}$$

The t_{sky} is equal to the sum of $t_{\Delta 1}$ and $t_{\Delta 2}$:

$$t_{sky} = (-\sqrt{2g(H_{to} - H_p)} - \sqrt{-2gH_p}) / g. \tag{A3}$$

With the known parameter t_{sky} and the motion frequency f it is easy to know the stand time and swing time. The velocity at the time of t_1 and t_2 are calculated based on free fall movement.