

ORIGINAL ARTICLE

Open Access



# Smart Gait: A Gait Optimization Framework for Hexapod Robots

Yunpeng Yin<sup>1</sup> , Feng Gao<sup>1\*</sup>, Qiao Sun<sup>1,2</sup>, Yue Zhao<sup>3</sup> and Yuguang Xiao<sup>1</sup>

## Abstract

The current gait planning for legged robots is mostly based on human presets, which cannot match the flexible characteristics of natural mammals. This paper proposes a gait optimization framework for hexapod robots called Smart Gait. Smart Gait contains three modules: swing leg trajectory optimization, gait period & duty optimization, and gait sequence optimization. The full dynamics of a single leg, and the centroid dynamics of the overall robot are considered in the respective modules. The Smart Gait not only helps the robot to decrease the energy consumption when in locomotion, mostly, it enables the hexapod robot to determine its gait pattern transitions based on its current state, instead of repeating the formalistic clock-set step cycles. Our Smart Gait framework allows the hexapod robot to behave nimbly as a living animal when in 3D movements for the first time. The Smart Gait framework combines offline and online optimizations without any fussy data-driven training procedures, and it can run efficiently on board in real-time after deployment. Various experiments are carried out on the hexapod robot *LittleStrong*. The results show that the energy consumption is reduced by 15.9% when in locomotion. Adaptive gait patterns can be generated spontaneously both in regular and challenge environments, and when facing external interferences.

**Keywords** Gait optimization, Swing trajectory optimization, Legged robot, Hexapod robot

## 1 Introduction

Legged robots have great off-road abilities. The quadrupeds like Spot [1], ANYmal [2, 3], HyQ [4] and MIT Cheetah series [5–8], and hexapods like Octopus[9] and Qingzhui [10, 11] showed major progress in locomotion control of legged robots, and demonstrated their applications in challenging terrains. However, there are still two deficiencies of legged robots. First, the legged robot requires repetitive limb motions to move around. This leads to high energy consumption. Second, human designed periodic gait patterns are stiff and rigid, unable to

achieve flexible behaviors as natural animals. Specifically, hexapods have more gait patterns due to the extra legs, predefined gaits could not fully leverage all the advantages of hexapods and limit their application in challenging environments. As these issues are related to the gait control of legged robots, here we propose the Smart Gait framework for hexapods, introducing three sub modules for gait optimization to address these deficiencies.

### 1.1 Related Works

The commonly used swing leg trajectories for legged robots are geometrically designed, like spline [12] or cycloid [13]. But these trajectories didn't consider the dynamics of the leg. MIT Cheetah [14] and its follower [15] used a Bezier curve for swing trajectory. They gave out the design principles but without the criteria in detail. Other energy based optimizations [16, 17] achieved success in swing trajectory design but they stopped to further discuss the relations between their results and the biological swing trajectories.

\*Correspondence:

Feng Gao  
gaofengsjtu@gmail.com

<sup>1</sup> State Key Laboratory of Mechanical System and Vibration,  
School of Mechanical Engineering, Shanghai Jiao Tong University,  
Shanghai 200240, China

<sup>2</sup> Lenovo Corporation, Shanghai, China

<sup>3</sup> AI Institute, School of Electronic Information and Electrical Engineering,  
Shanghai Jiao Tong University, Shanghai 200240, China



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence 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 licence, visit <http://creativecommons.org/licenses/by/4.0/>.

For gait patterns, formerly we have summarized that the relations between gait parameters and travel speed in consideration of energy expenditures [18], and also demonstrated several gait adaptation methods on irregular terrains [11, 19] or in fault-tolerant cases [20]. Central Pattern Generators (CPGs) [21–23] are powerful tools to generate different gait patterns but still limited to periodic movement. To make up for this, Feasible Impulse Set (FIS) [24] was first proposed on MIT Cheetah 3 robot. By considering the impulse that can be delivered by each leg, it coordinated adaptive lift-off and touch-down events of the legs without preset gait patterns. However, due to the computing complexity of the convex set, it can only response to single dimensional movements with fixed leg swing time. Our original idea of Smart Gait was inspired by this scheme and we further developed it. Our Smart Gait can handle 3D locomotion with adaptive swing time.

Gait planning in 3D environments is important for legged robots. Free Gait [25] or TOWR [26] are good frameworks for route planning, which can generate both body and swing leg trajectories under the constrain of preset gait parameters and terrain features. Similar research is carried out on a hexapod [27] using quasi-static gaits. By setting a target point, the robot can exert its traffability after the whole-body planning is solved. However, they didn't talk about unexpected external interferences on the robot. Differently, our Smart Gait framework is based on online state of the robot but not targeting on a final point. It concentrating on the energy consumption and the biological flexibility, that generates the gaits without preset parameters instructed by human. More specifically, Smart Gait decides whether a certain leg in contact stage should lift or not at each moment, rather than repeating obstinate step cycles before the final target. Moreover, as Smart Gait is online state based, it can handle external interference as well.

## 1.2 Contributions

Legged animals can generate vivid gait patterns or swing leg trajectories spontaneously based on their motion states or terrains without precise clock or trajectories in their organisms. To catch up with these natural animals' characteristics, here we propose the Smart Gait framework, making it possible for the hexapod robot to decide its own gaits and swing leg trajectories based on its online motion states or terrains. We propose three modules of gait optimization progressively in our Smart Gait framework. The contributions of Smart Gait framework are summarized as follows.

(A). Swing trajectory optimization. By considering the full dynamics of the leg linkages and the impulse loss, it reduces the energy consumption and iner-

tial counterforce on the fuselage during the swing phase. To our knowledge, we are the first to generate a swing trajectory conforming to the biological one.

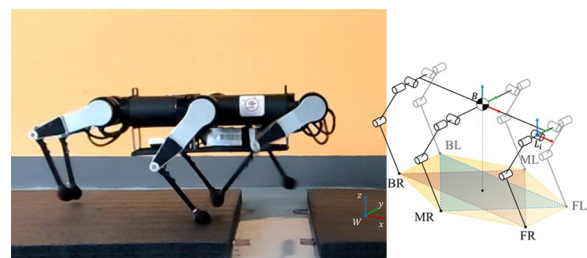
(B). Period & Duty optimization. By considering the swing trajectory loss and the centroid dynamics of the robot, it can adjust the swing and contact time according to different travel speeds. It helps to reduce energy consumption and leads to a human intuition consistent gait pattern.

(C). Gait sequence optimization. By considering the swing legs loss and centroid dynamics of the robot, along with the body balance and leg workspace loss, it allows the robot itself to decide leg lifting timing according to its online state, to obtain the elegance and flexibility as a living animal. To our knowledge, we are the first to develop this idea in 3D motion of legged robots in real time.

This paper is constructed as follows. Section 2 introduces our hexapod robot *LittleStrong*. Section 3 proposes the three modules of our Smart Gait framework progressively. Section 4 carries out five experiments to verify the feasibility of the sub-modules of the Smart Gait framework. And lastly Section 5 draws the conclusions.

## 2 Robot Overview

The bionic designed *LittleStrong* [28] is a hexapod robot inspired by insects as shown in Figure 1. Its six legs are symmetrically distributed on both sides of the body, named as Back-Right leg (BR), Back-Left leg (BL), Middle-Left leg (ML), Front-Left leg (FL), Front-Right leg (FR), Middle-Right leg (MR), respectively. In order to avoid leg collisions during movement, the middle legs are installed a little farther from the sagittal plane. The robot is about 0.4 m in height, 0.68 m in length and 0.46 m in width when standing. For each leg, there are 3 DoFs, serially constructed as hip, thigh and shank joints. Therefore, *LittleStrong* has 18 joints in total. Each joint is driven by a brushless electric motor with a planetary gear set and an embedded encoder. The onboard computer adopts the GNU/Linux with the Xenomai [29] real-time patch,



**Figure 1** The hexapod robot *LittleStrong*, photograph (left), configuration (right), and coordinate systems

communicating with 18 drivers and an Inertial Measurement Unit (IMU) through EtherCAT protocol, and getting remote commands through ZeroMQ [30]. It runs a Whole-Body Control (WBC) [31] algorithm for motion control.

The coordinate systems are also defined in Figure 1. The World Coordinate System (WCS) is fixed with the ground, where the most locomotion planning is in. The Body Coordinate System (BCS) is fixed at the center of its body, and all variables listed in BCS will be marked with a left superscript {B}. Each leg has its own Leg Coordinate System (LCS), which is fixed with the BCS but takes the particular leg’s hip joint as its origin. All variables in the LCS will be marked with a left superscript {L}. The right subscript {i} is for the leg number in the following sections.

### 3 Gait Optimization

The architecture of Smart Gait is shown in Figure 2. The offline swing leg trajectory optimization (module A) and gait period & duty optimization (module B) parameterize the results and store them for online call. The online Gait Adaptor (module C) takes the real-time state of the robot for the gait sequence optimization. Gait sequence contains a signal that shows whether a certain leg in contact phase should be lifted. The Gait Scheduler uses the optimized gait period & duty and the sequence signal to generate the gait rhythm. After that, the gait rhythm and the reference swing trajectories are handed to WBC [31] to generate the joint commands for the motors.

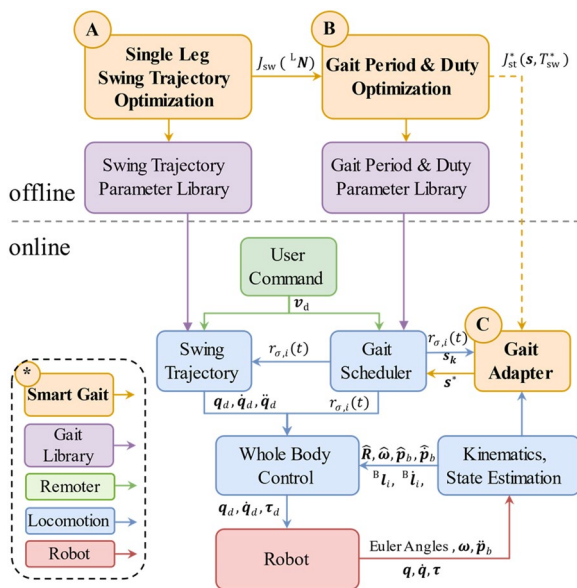


Figure 2 Architecture of Smart Gait framework

### 3.1 Trajectory Optimization for Swing Legs

The swing leg trajectory is the curve connecting the lifting point and the landing point of the swing leg’s tip. Traditionally, there are cycloid or symmetric polynomial trajectories. These trajectories are constructed based on simple geometric features and are convenient to use. However, without considering the dynamic characteristics, the movements of the swing leg would consume more energy and have greater impact on the fuselage due to the inertial reaction forces brought by unpleasant accelerations and decelerations.

Here, a two-segmented 4th order Bezier curve is used for the swing trajectory. First, it has enough optimization intervals without being too complicated. Second, when approaching the middle time of swing process, the junction of two sub-Bessel curves helps to reach the preset step height. The swing trajectory is a 3D spatial curve, and the optimization is carried out in the LCS. Figure 3 shows the two segments of the 4th order Bezier curve for the right leg (red and blue thick solid line) and its control nodes (red and blue solid points  $\{{}^L N_1 \sim {}^L N_9\}$ ). The thin lines are the leg links that tracking over the trajectory. The expression of each segmented Bezier curve is:

$${}^L p_{sw}(t) = \sum_{k=0}^n C_n^k \cdot (1 - r_{\sigma}(t))^{n-k} \cdot (r_{\sigma}(t))^k \cdot {}^L N_{k+\sigma}, \tag{1}$$

$$C_n^k = \frac{n!}{k!(n-k)!}, C_n^0 = 1,$$

$$\sigma = \begin{cases} 0, & t < T_{sw}/2, \\ 5, & t \geq T_{sw}/2, \end{cases}$$

$$r_{\sigma}(t) = \begin{cases} 2t/T_{sw}, & t < T_{sw}/2, \\ 2t/T_{sw} - 1, & t \geq T_{sw}/2, \end{cases}$$

where  $n = 4$ ,  ${}^L N_{k+\sigma}$  is the  $(k+m)$ th control node of Bezier curve; is the combinatorial number;  $T_{sw}$  is the swing duration;  $r_{\sigma}(t)[0,1]$  is the gait rhythm, a normalized

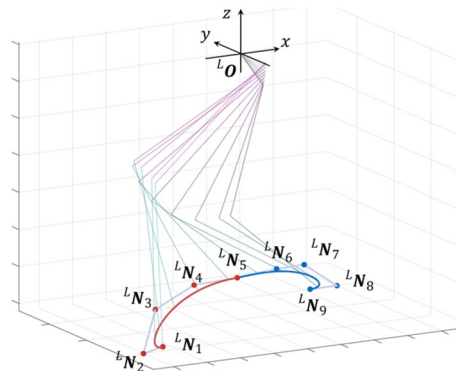


Figure 3 Segmented 4th order Bezier curve to be optimized in LCS

mapping of swing time  $t$ . The subscript  $\{\sigma\}$  is used to distinguish the two segments of the Bezier curve.

According to the leg swing trajectory  ${}^L\mathbf{p}_{sw}(t)$ , the generalized force  $\boldsymbol{\tau}(t)$  is calculated from the inverse kinematics and the floating base dynamics of the leg:

$$\mathbf{q}_j(t) = \mathbf{IK} \left( {}^L\mathbf{p}_{sw}(t) \right), \tag{2}$$

$$\dot{\mathbf{q}}_j(t) = \mathbf{J}_v^{-1}(\mathbf{q}_j(t)) \cdot {}^L\dot{\mathbf{p}}_{sw}(t), \tag{3}$$

$$\ddot{\mathbf{q}}_j(t) = \dot{\mathbf{J}}_v^{-1}(\mathbf{q}_j(t)) \cdot {}^L\dot{\mathbf{p}}_{sw}(t) + \mathbf{J}_v^{-1}(\mathbf{q}_j(t)) \cdot {}^L\ddot{\mathbf{p}}_{sw}(t), \tag{4}$$

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t)) \cdot \ddot{\mathbf{q}}(t) + \mathbf{h}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)), \tag{5}$$

where  $\mathbf{q}_j(t) \in \mathbb{R}^3$  is the vector of joint angles, obtained from the leg's inverse kinematics  $\mathbf{IK}({}^L\mathbf{p}_{sw}(t))$ .  $\mathbf{J}_v(\mathbf{q}_j(t))$  is the leg's Jacobian.  $\mathbf{q}(t) = [{}^L\mathbf{q}_{FB}^T \ \mathbf{q}_j^T(t)]^T$  is a generalized vector, in which  ${}^L\mathbf{q}_{FB} \in \mathbb{R}^6$  is the generalized coordinate (position and rotation) of the floating base for the leg. As this swing trajectory is expressed in the LCS, thus  ${}^L\mathbf{q}_{FB} \equiv 0$  (the floating base never moves in its own coordinate system).  $\mathbf{M}(\mathbf{q}(t)) \in \mathbb{R}^{9 \times 9}$  is the generalized inertial matrix;  $\mathbf{h}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) \in \mathbb{R}^9$  is the vector for the nonlinear terms;  $\mathbf{g}(\mathbf{q}(t)) \in \mathbb{R}^9$  is for the gravity;  $\boldsymbol{\tau}(t) = [\boldsymbol{\tau}_{FB}^T(t) \ \boldsymbol{\tau}_j^T(t)]^T$  is the vector of generalized forces/torques, where  $\boldsymbol{\tau}_{FB}(t) \in \mathbb{R}^6$  is for the floating base and  $\boldsymbol{\tau}_j(t) \in \mathbb{R}^3$  is for the joints. Namely,  $\boldsymbol{\tau}_{FB}(t)$  is the vector of reaction force / torque of the leg on the fuselage.

Then, by building and minimizing the cost function that tracking the swing trajectory, the control nodes of Bessel curve can be obtained as:

$${}^L\mathbf{N}^* = \arg \min_N J_{sw}({}^L\mathbf{N}), \tag{6}$$

$$J_{sw}({}^L\mathbf{N}) = \int_0^{T_{sw}} \left( \alpha \|\boldsymbol{\tau}_j(t)\|^2 + \beta \boldsymbol{\tau}_j^T(t) \cdot \dot{\mathbf{q}}(t) + \gamma \|\boldsymbol{\tau}_{FB}(t)\|^2 \right) dt, \tag{7}$$

$$\text{s.t.,} \begin{cases} {}^L\mathbf{p}_{sw}(0) = {}^L\mathbf{p}_0, \\ {}^L\mathbf{p}_{sw}(t_f) = {}^L\mathbf{p}_f, \\ {}^L\dot{\mathbf{p}}_{sw}(0) = {}^L\mathbf{v}_d, \\ {}^L\dot{\mathbf{p}}_{sw}(t_f) = {}^L\mathbf{v}_d, \\ z({}^L\mathbf{p}_{sw}(t_m)) = -h_{body} + h_{step}, \\ {}^L\dot{\mathbf{p}}_{sw}(t_{m-}) = {}^L\dot{\mathbf{p}}_{sw}(t_{m+}), \end{cases} \tag{8}$$

$$\begin{cases} {}^L\mathbf{p}_f = -{}^L\mathbf{p}_0 = \left[ \frac{v_x T_{sw}}{2} \ \frac{v_y T_{sw}}{2} \ -h_{body} \right]^T, \\ {}^L\mathbf{v}_d = \left[ -v_x \ -v_y \ 0 \right]^T, \end{cases} \tag{9}$$

where the cost function of Eq. (7) is composed of three integral items: equivalent electrical power loss (joint

torque is proportional to the driver's current), mechanical power loss, and body impact reaction force during the swing phase.  $\{\alpha, \beta, \gamma\}$  are their corresponding weights. The swing trajectory shall satisfy the constraints in Eq. (8), namely starting position  ${}^L\mathbf{p}_0$ , ending position  ${}^L\mathbf{p}_f$ , starting and ending velocity  ${}^L\mathbf{v}_d$ , midpoint height, and midpoint velocity continuity. As  $h_{body}$  is the height of the robot body;  $h_{step}$  is the step height;  $\{v_x, v_y\}$  are the translational velocity of the robot in horizontal directions. This nonlinear optimization equation is solved by LM [32] method. After obtaining the control nodes, the segmented Bessel curve, namely the swing trajectory, can be reconstructed as Eq. (1).

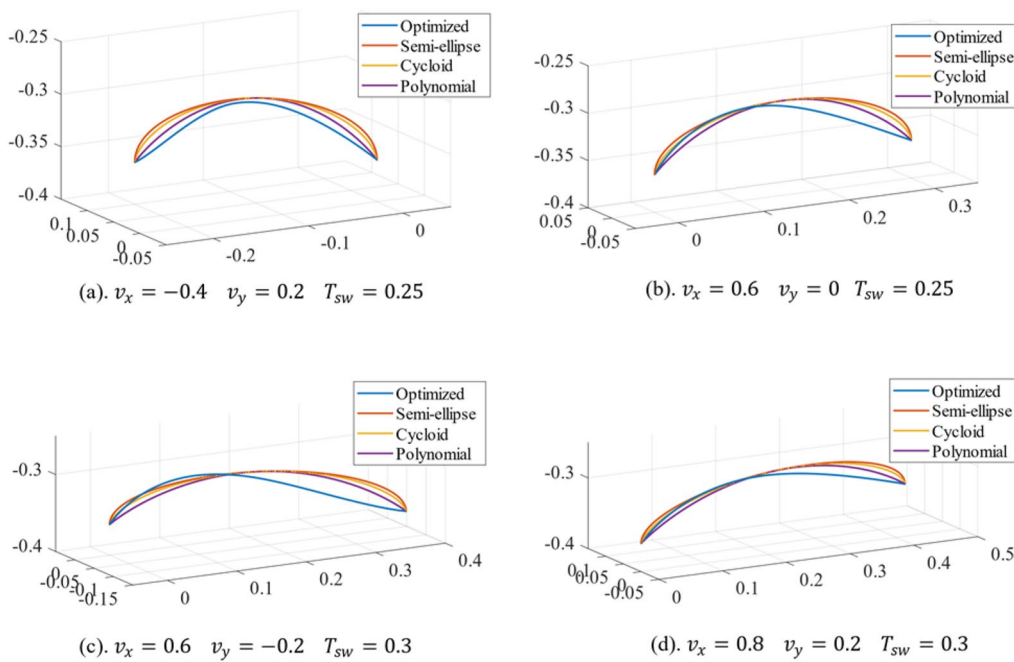
The above process is carried out offline. We store the working condition parameters and build a swing trajectory library, and then it can be called online to get the control nodes for the swing trajectory due to different translational velocities of the robot. Since the locomotion control is carried out in the WCS, the transformation of the swing trajectory from the LCS to the WCS can be expressed as:

$$\mathbf{p}_{sw,i}(t) = \mathbf{p}_{body}(t) + \mathbf{R}(t) \left( {}^B\mathbf{p}_{hip,i} + {}^L\mathbf{p}_{sw}(t) \right), \tag{10}$$

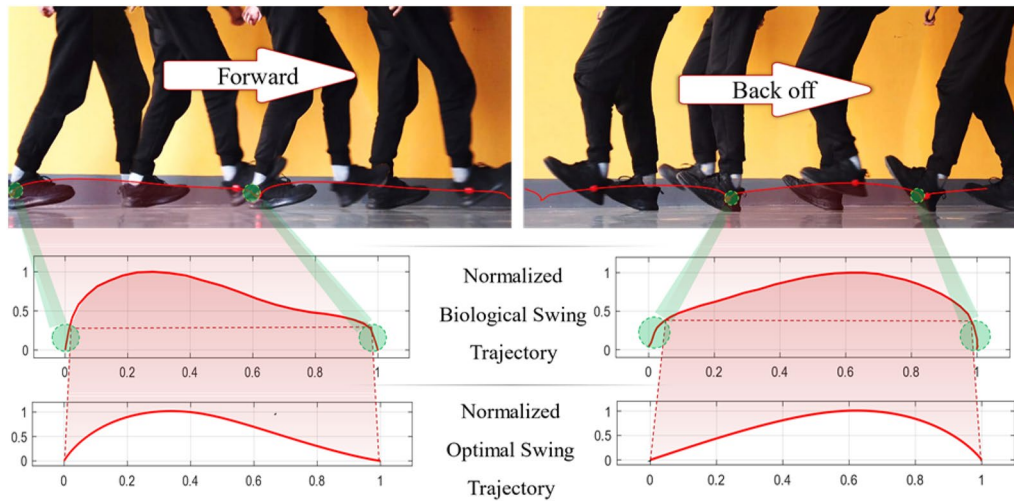
where  $\mathbf{p}_{body}$  and  $\mathbf{R}$  represent the position and rotation of the robot,  ${}^B\mathbf{p}_{hip,i}$  is the  $i$ th hip position relative to the body,  ${}^L\mathbf{p}_{sw}(t)$  is the  $i$ th leg swing trajectory under its own LCS.

Figure 4 shows the comparisons of several groups of different swing trajectories in the WCS. As cycloid and symmetric cubic polynomial are all curves constructed based on simple geometric features, they are all in the same vertical plane coinciding the starting point and the end point. While the optimized trajectory is a spatial curve not in this plane.

In addition, as shown in Figure 5, when the knee bending direction is consistent with the travel direction, the first half of the optimized swing trajectory is higher than the second half; and when the knee's bending direction is opposite to the travel direction, the second half of the of the trajectory is higher than the first half. This is consistent with the corresponding relationship between the biological configurations and swing trajectories. In Figure 5, we track the swing trajectories with a red tag fixed on the side of the experimenter's shoes when moving forward (knee forward) and moving back off (knee opposite), respectively. The experimenter only uses the heel to walk in order to match the configuration of the robot's leg (a three-linkage-model with hip, thigh and shank). Regardless of the compression and rebound caused by the elasticity of the shoes' soles (circled in green), our optimized swing trajectories is very consistent with the trend of the biological ones. We conducted experiments on the robot



**Figure 4** Energy cost of Period & Duty at different given speeds (The flat area will exceed the leg workspace thus not considered)



**Figure 5** Comparison between biological swing trajectories and optimized swing trajectories

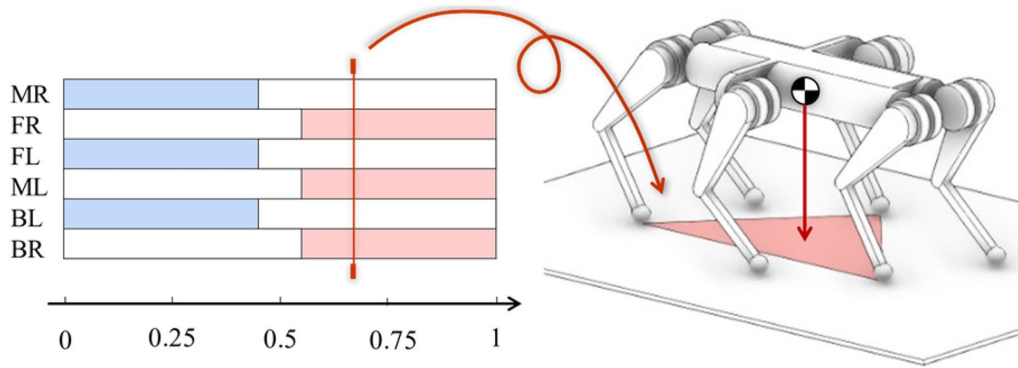
in Section 4.1. It proved that our optimized trajectory has lower energy consumption and softer reaction force to the fuselage.

### 3.2 Period & Duty Optimization for Gait Cycles

The gait cycle of each leg is determined by three parameters: period, duty, and phase-difference when leaving contact phase to swing phase. The period  $T$  is the time duration for a complete step (swing time + contact time,  $T = T_{sw} + T_c$ ). Duty  $\lambda$  is the percentage of the contact

phase to the period (contact time/period,  $\lambda = T_c/T$ ). The phase-difference between different legs determines the gait patterns. Figure 6 shows the Tripod-trot [33] gait pattern commonly used by hexapod robots.

When in a given moving speed, different gait parameters will greatly affect the energy consumption of the robot. This subsection will optimize the gait period & duty under the tripod-trot gait pattern (fixed phase-differences). Phase-differences optimization will be discussed in next subsection.



**Figure 6** Tripod-trot gait pattern for hexapods (The white in the sequence diagram represents the swing phase; the red and blue represent the contact phase, respectively)

For the swing legs, the energy consumption comes from the motion of the leg, namely the dynamic items as calculated in Eq. (7). But for the energy consumption of contact legs, it contains two parts. One part is for the motion of the leg itself, and another part is the support force for the body. Both leg-dynamics and body-support items need to be considered at the same time. When the robot moves with a given speed, in the BCS, the trajectory  ${}^B\mathbf{l}_i(t)$  of the contact leg is considered as a uniform line that connects the foot-landing point to the foot-lifting point of next step cycle (regardless of turning). The leg-dynamics subitem of the energy consumption for the contact leg can be obtained in the same way as in Eq. (5). For the body-support subitem, we get the joints' torques by distributing body gravity to each contact leg with a simplified Centroid Balance Control (CBC)[34] method.

$$\begin{cases} \sum_1^{c(t)} J_{v,i}^{-T}(t) \cdot \boldsymbol{\tau}_{c,i}(t) = m\mathbf{g}, \\ \sum_1^{c(t)} {}^B\mathbf{l}_i(t) \times \left( J_{v,i}^{-T}(t) \cdot \boldsymbol{\tau}_{c,i}(t) \right) = \mathbf{0}, \end{cases} \quad (11)$$

$$\underbrace{\begin{bmatrix} J_{v,1}^{-T}(t) & \dots & J_{v,6}^{-T}(t) \\ [{}^B\mathbf{l}_1(t)]_{\times} J_{v,1}^{-T}(t) & \dots & [{}^B\mathbf{l}_6(t)]_{\times} J_{v,6}^{-T}(t) \end{bmatrix}}_{A(t)} \underbrace{\begin{bmatrix} \boldsymbol{\tau}_{c,1}(t) \\ \vdots \\ \boldsymbol{\tau}_{c,6}(t) \end{bmatrix}}_{\boldsymbol{\tau}_c(t)} = \underbrace{\begin{bmatrix} m\mathbf{g} \\ 0 \end{bmatrix}}_b, \quad (12)$$

where  $m$  is the mass of the robot and  $\mathbf{g}$  stands for the gravity.  $J_{v,i}(t)$  is the Jacobian matrix for each leg;  $c(t)$  represents the legs in contact phase;  $\boldsymbol{\tau}_c(t)$  is used to represent the supporting torques of the contact leg;  ${}^B\mathbf{l}_i(t)$  is the  $i$ th leg tip position in BCS. For the simplification of the optimization problem, rotation is not considered in the offline optimization. Eq. (11) is rewritten to Eq. (12) as the matrix form.  $[*]_{\times}$  represents an antisymmetric matrix

operator. Since the contact state of each leg is constantly changing during walking, the torques of contact legs' joints are distributed by solving a quadratic programming (QP) problem as:

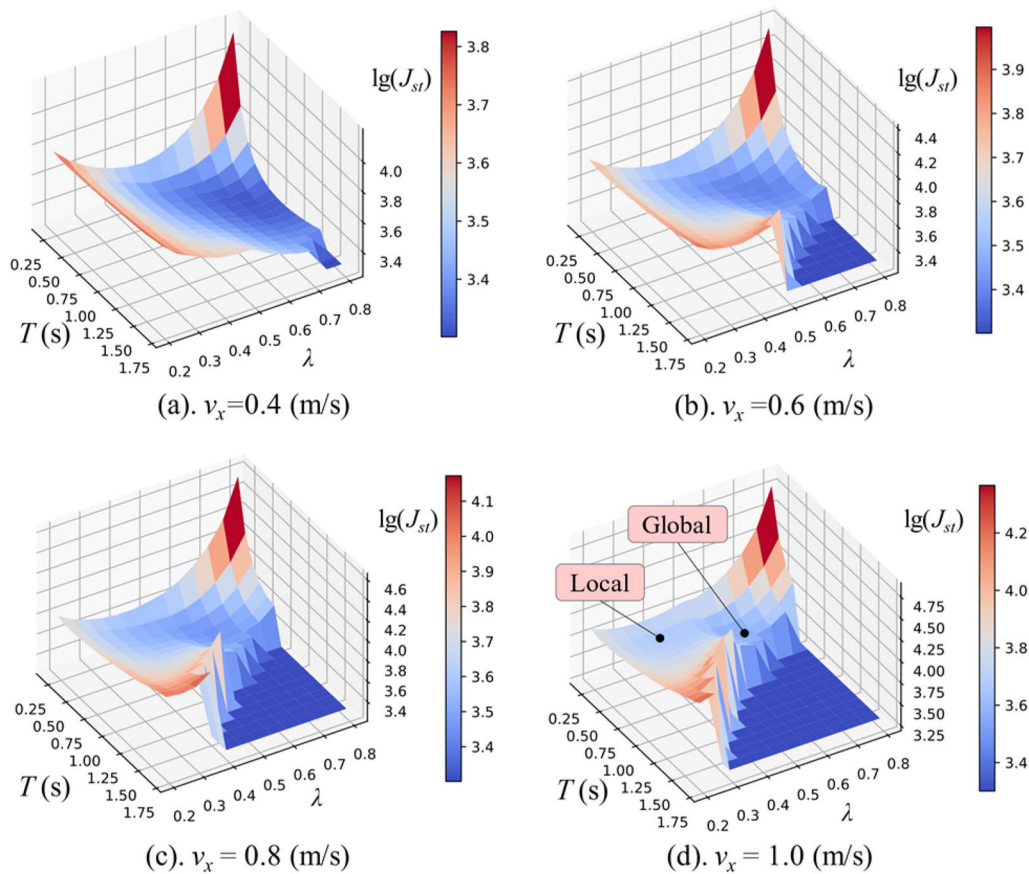
$$\begin{aligned} \boldsymbol{\tau}_c^*(t) &= \arg \min_{\boldsymbol{\tau}_c \in \mathbb{R}^{18}} \left( \|A(t) \cdot \boldsymbol{\tau}_c(t) - \mathbf{b}\|_W^2 + \|\boldsymbol{\tau}_c(t)\|^2 \right), \\ \text{s.t.}, \quad &\begin{cases} C(t) \cdot J_{v,i}^{-T}(t) \cdot \boldsymbol{\tau}_{c,i}(t) \leq f_{\text{lim},i}, \\ |\boldsymbol{\tau}_c(t)| \leq \boldsymbol{\tau}_{\text{max}}, \\ f_{\text{lim},i} = \begin{cases} f_{\text{lim}}, & \text{leg } i \text{ in stance,} \\ 0, & \text{leg } i \text{ in swing,} \end{cases} \end{cases} \end{aligned} \quad (13)$$

where  $W$  is the weight matrix of the quadratic term, and the inequality constraints represent the friction cone [34] and the extreme values of joint torques respectively. After getting the joint torques  $\boldsymbol{\tau}_c^*(t)$ , we build the energy cost of walking process as:

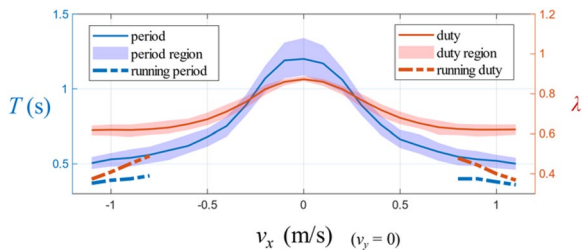
$$\begin{aligned} J_{\text{st}}(v, T, \lambda) &= 6 \cdot J_{\text{sw}}({}^L\mathbf{N}^*) \\ &+ \int_0^T \left( \alpha \|\boldsymbol{\tau}_{J,c}(t) + \boldsymbol{\tau}_c^*(t)\|^2 + \beta (\boldsymbol{\tau}_{J,c}(t) + \boldsymbol{\tau}_c^*(t))^T \cdot \dot{\mathbf{q}}(t) \right) dt. \end{aligned} \quad (14)$$

The single leg's swing phase cost  $J_{\text{sw}}({}^L\mathbf{N}^*)$  is obtained from the previous subsection, and it needs to multiply the number of legs by "6". The second integral item is similarly combined as equivalent electrical power cost and mechanical power cost.  $\boldsymbol{\tau}_{J,c}(t)$  is the torque for leg-dynamics, obtained in the same way as Eq. (5);  $\boldsymbol{\tau}_c^*(t)$  is body support torques as previously defined.

Figure 7 shows how the period & duty parameters influence on energy cost at different given speeds of the robot. Note that the cost value is displayed in logarithmic coordinates. The flat area at the lower right corner of each figure represents that this Period & Duty combination will exceed the leg workspace and cannot be considered.



**Figure 7** Energy cost of Period & Duty at different given speeds (The flat area will exceed the leg workspace thus not considered)



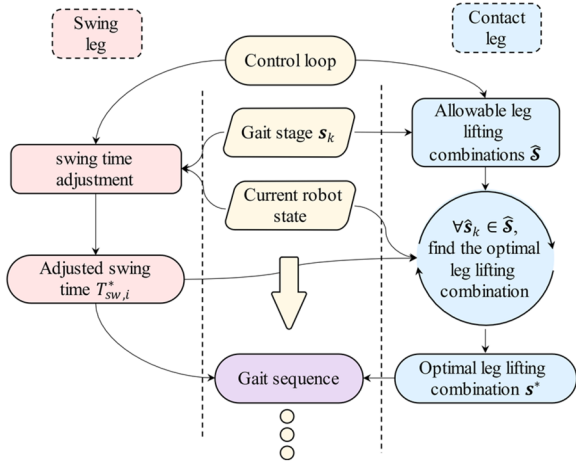
**Figure 8** Relations between speed and optimal gait period & duty

Figure 8 shows the relationship between the optimal period & duty at different speeds. Back in Figure 7, we notice that the gradient around the optimal period & duty is very small. Therefore, we add an adjustment region in Figure 8, in which the optimal cost is allowed to fluctuate by 1%. When in a lower speed, the allowed period floating range is large, but as the duty is already at a large level, the duty floating range is small. As the target speed increases, the period and duty tend to be unchanged, but

from Figure 7, there appears another local minima in  $\lambda < 0.5$  area, which corresponds to a running gait with flying phase. This is completely consistent with our human's intuition that when the speed increases to a certain stage, the walking gait should be replaced by the running gait. However, for the high-speed conditions, the workspace limitation will become a major constraint factor, and the adjustments of period & duty will be greatly limited. Therefore, this paper will not discuss the high-speed gait ( $v > 1$  m/s and  $\lambda < 0.5$ ). Relevant comparative experiments are given in Section 4.2.

### 3.3 Gait Sequence Optimization for Leg Lifting Events

The phase-difference of each leg is determined by the timing that lifting this leg from contact stage to swing stage. When should the lifting event be taken is the gait sequence concerned. In fact, it is hard to build a gait sequence library like previous modules due to the enlarged state parameters scale, and it is difficult to cover the potential online states of the robot either. So, we here adopt the online Gait Adapter as shown in Figure 9,



**Figure 9** Scheme of online Gait Adapter for gait sequence optimization

which can process the 3D motion of the robot to generate the optimal leg lifting sequence.

We define a set  $\mathcal{S}$  to represent the combination of the legs' current stages:

$$\mathcal{S} = \left\{ \mathbf{s} = [s_1 \cdots s_6]^T \mid s_i = \begin{cases} 1, C_1, \\ 0, C_2, \end{cases} i \in \{1, \dots, 6\} \right\}, \quad (15)$$

$$\begin{cases} C_1 : \text{leg } i \text{ in swing stage,} \\ C_2 : \text{leg } i \text{ in contact stage.} \end{cases}$$

Then the next action of the robot (ideal combination of legs' stages in the next moment) should be determined by the current robot state:

$$\mathbf{s}_{k+1} \sim \left( \mathbf{p}_{\text{body}}, \mathbf{R}, \mathbf{v}_{\text{body}}, \boldsymbol{\omega}_{\text{body}}, \mathbf{p}_{\text{foot}}, T_{\text{sw}}^*, \mathbf{s}_k \right), \quad (16)$$

$$\mathbf{s}_k, \mathbf{s}_{k+1} \in \mathcal{S}.$$

That is, which leg should enter the swing stage in the next moment is related to the robot's position, rotation, velocity, angular velocity, feet positions, legs swing time and legs stages at present.

For the swing leg, there is a fact that for natural myriapod animals, each leg's swing time is not fixed and even varies between legs in a single step cycle according to different motion states. Encouraged by this fact, we adjust each leg's swing time by the current state of the robot. The benchmark of the leg's swing time is obtained from the previous section (based on tripod-trot gait). Although the optimized leg lifting sequence may not conform to the beat of tripod-trot, but considering that the swing phase only takes a small proportion in the walking energy cost in Eq. (14), this does not affect the

optimization result too much. Here we design the relationship between the leg swing time and the balance state of the fuselage. When the robot tends to topple to one side or fails to reach the preset state, we shorten the leg swing time on the corresponding side, then this swing leg can enter contact phase in time to correct the body balance error:

$$T_{\text{sw},i}^* = T_{\text{sw}} - \Delta t_i^*, \quad (17)$$

$$\Delta t_i^* = \begin{cases} 0, & \Delta t_i \leq 0, \\ \Delta t_i, & 0 < \Delta t_i \leq \Delta t_{\text{max}}, \\ \Delta t_{\text{max}}, & \Delta t_{\text{max}} < \Delta t_i, \end{cases}$$

$${}^B \dot{\boldsymbol{\omega}} = \mathbf{K}_P \left( \hat{\mathbf{R}}^T \log \left( \mathbf{R}_d \hat{\mathbf{R}}^T \right) \right) + \mathbf{K}_D \left( {}^B \boldsymbol{\omega}_d - {}^B \hat{\boldsymbol{\omega}} \right), \quad (18)$$

$${}^B \mathbf{n} = \begin{bmatrix} \frac{{}^B v_x}{\|v\|} & \frac{{}^B v_y}{\|v\|} & 1 \end{bmatrix}^T, \quad (19)$$

$$\Delta t_i = \left( {}^B \mathbf{l}_i \times {}^B \mathbf{n} \right) {}^B \dot{\boldsymbol{\omega}}, \quad (20)$$

where  $T_{\text{sw}}$  is the benchmark of leg swing time obtained in the previous subsection;  $\Delta t_i^*$  is the adjust time;  $\Delta t_{\text{max}}$  is the maximum allowable adjustment;  ${}^B \dot{\boldsymbol{\omega}}$  is the attitude correction vector derived from a PD control;  $\mathbf{R}_d$  and  ${}^B \boldsymbol{\omega}_d$  are the body target posture and angular velocity;  $\hat{\mathbf{R}}$  and  ${}^B \hat{\boldsymbol{\omega}}$  are the estimated body posture and angular velocity. The  $\log^*$  operator maps  $\text{SO}(3)$  to  $\mathbb{R}^3$ .  $\{\mathbf{K}_P, \mathbf{K}_D\}$  are positive definite diagonal matrices of corresponding weights, respectively;  ${}^B \mathbf{n}$  is the generalized target velocity direction and  ${}^B \mathbf{l}_i$  is for the tip position of the  $i$ th leg in the BCS. Then the swing time adjustment  $\Delta t_i$  reflects how much earlier the  $i$ th swing leg needs to enter contact phase, so that can generate more driving force for the correction vector to keep the body balance.

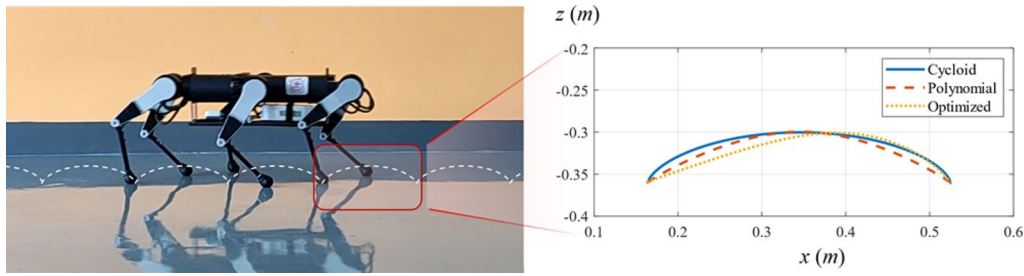
For the contact leg, we use the idea of model predictive control for the decisions of leg lifting timing for the gait sequence. At each moment, all the potential leg stage combinations are traversed to find the best combination for execution. The cost function of leg stage combination  $\mathbf{s}$  is as follows:

$$J_s(\mathbf{s}) = J_{\text{st}}^*(\mathbf{s}, T_{\text{sw}}^*) + \int_0^{T_h} E(t) dt, \quad (21)$$

$$E(t) = \|\dot{\mathbf{e}}_b(t)\|_{Q_a}^2 + \|\dot{\mathbf{e}}_b(t)\|_{Q_v}^2 + \|\mathbf{e}_b(t)\|_{Q_p}^2 + \sum_i^{c(t)} \left\| \int \mathbf{e}_{l,i}(t) \right\|_{Q_l}^2,$$

$$\ddot{\mathbf{e}}_b(t) = \mathbf{A}(t)\boldsymbol{\tau}(t) - \mathbf{b},$$





**Figure 10** Robot moves with different swing trajectories

$$\dot{e}_b = \int_0^{T_h} \ddot{e}_b(t)dt,$$

$$e_b = \int_0^{T_h} \dot{e}_b(t)dt,$$

$${}^L e_{l,i}(t) = \begin{cases} 0, & {}^L l_i(t) \leq {}^L l_{lim}, \\ {}^L l_i(t) - {}^L l_{lim}, & {}^L l_i(t) > {}^L l_{lim}, \end{cases}$$

where the first item  $J_{st}^*(s, T_{sw}^*)$  is a function similar to Eq. (14), but the target leg stage combination  $s$  is used instead of a fixed Tripod-Trot gait pattern, and the swing time  $T_{sw}^*$  mentioned above is adopted.  $T_h$  is the prediction horizon. Usually,  $T_h$  should be set greater than one step period ( $T_h \geq T_{sw} + T_c$ ). The second integral item is to predict the cost of body balance and leg workspace.  $\ddot{e}_b(t)$  is the predictive generalized acceleration deviation of body balance after Eq. (12). Correspondingly,  $\dot{e}_b(t)$  and  $e_b(t)$  are the generalized velocity and position deviation. And  $\{Q_a, Q_v, Q_p\}$  are their weighting matrices.  ${}^L e_{l,i}(t)$  is the deviation of the predicted tip position of each contact leg that beyond the workspace constraint in the LCS, with  ${}^L l_i(t)$  for the tip position and  ${}^L l_{lim}$  for the leg workspace.  $Q_l$  is the corresponding weighting matrix.

Considering that not all leg stage combinations are reasonable for execution when the robot is running, a set  $\hat{S}$  is defined to represent the allowable leg lifting combinations due to the current leg stages for each moment:

$$\hat{S} := \left\{ \hat{s}_k \in \mathcal{S} \mid \hat{s}_{k,i} = \begin{cases} 1, \hat{C}_1, \\ 0, \hat{C}_2, \end{cases} i \in \{1, \dots, 6\} \right\}, \quad (22)$$

$$\begin{cases} \hat{C}_1: \text{leg } i \text{ in swing stage, or ready to swing,} \\ \hat{C}_2: \text{leg } i \text{ in contact stage, and will hold still.} \end{cases}$$

By traversing  $\forall \hat{s}_k \in \hat{S}$ , we get the optimal leg lifting combination as:

$$s_c^* = \arg \min_s J_s(\hat{s}_k). \quad (23)$$

Now,  $s_c^*$  is just the optimal leg stage under the current state, whether it is the optimal stage in the

full-time-domain can be further compared with the next forecast moment:

$$s^* = \begin{cases} s_c^*, & J_s(s_c^*) < J_s^+(s_c^*), \\ s_c, & J_s(s_c^*) \geq J_s^+(s_c^*). \end{cases} \quad (24)$$

$J_s^+(s_c^*)$  represents the cost value that keep still the leg stage  $s_c$  at the current moment but lifting the leg according to  $s_c^*$  in the next forecast moment. If the cost of  $J_s(s_c^*)$  is less than  $J_s^+(s_c^*)$ , it means that  $s_c^*$  is the optimal leg lifting selection and can be executed; otherwise, maintain the current leg stage  $s_c$ .

The three modules of Smart-Gait framework are fully established. Experiments are taken in the next section.

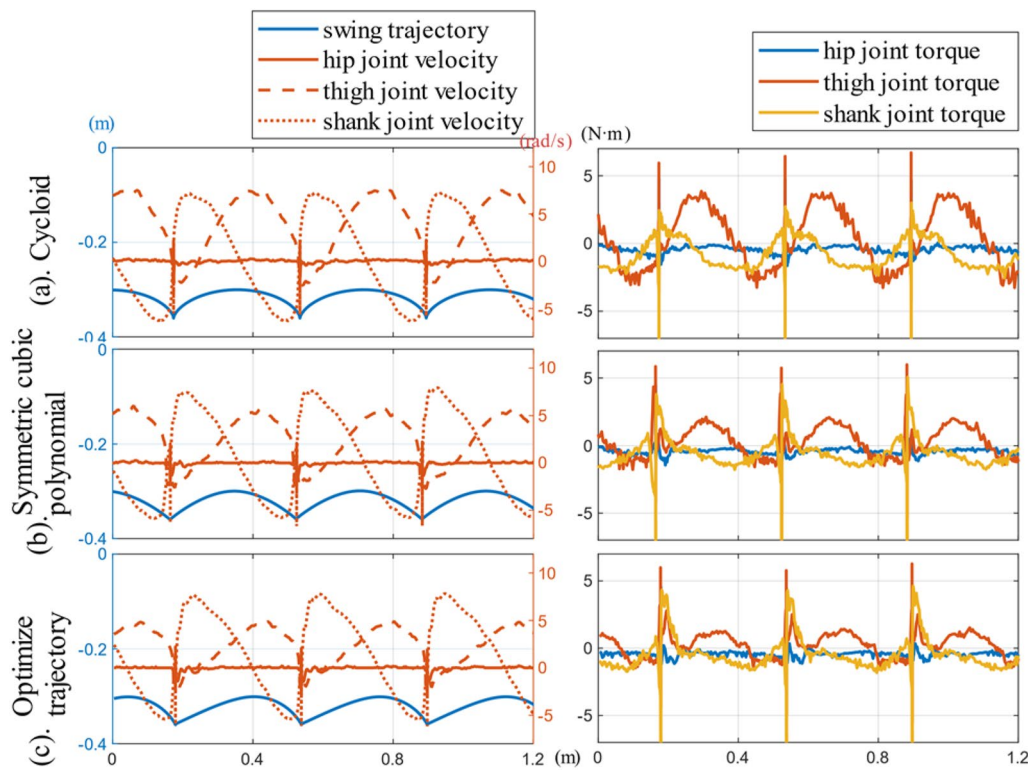
## 4 Experiments and Results

We conducted several experiments on *LittleStrong* to verify each module of our Smart Gait framework. Experiment (4.1) is to verify the energy consumption of the optimized swing trajectory (Additional file 1); Experiment (4.2) is to verify the outcome of Period & Duty optimization in tripod-trot gait (Additional file 2). Experiments (4.3) (Additional file 3), Experiments (4.4) (Additional file 4), Experiments (4.5) (Additional file 5) are to verify the gait transitions of the online Gait Adapter.

### 4.1 Swing Trajectories Comparison

This experiment is to verify the energy consumption of the optimized swing trajectory. The robot moves with different swing leg trajectories under the same gait pattern, as Figure 10 shows. We compared the energy consumption between cycloid, symmetric cubic polynomial, and our optimized swing trajectories.

As shown in Figure 11, the relevant test data of the robot's back-right leg is shown. The joints velocities and torques of the cycloidal trajectory are the largest, followed by the symmetric cubic polynomial trajectory, and the joints velocities and torques of our optimized trajectory are the smallest. In addition, the angular velocity changing amplitude of the optimized trajectory is also the smallest. That is, the required acceleration



**Figure 11** Joint velocities and torques of the Back-Right leg of the robot under different swing trajectories (The horizontal axis is the tip position; thus, the contact phase will be compressed)

and deceleration are gentler, which reduces the inertial impact force on the fuselage, and benefits to the stability of the robot.

Since the electric system’s impedance parameters are not obtained, we compare them with normalized relative energy consumption (The average of the current magnitude of the swing leg’s motors). As shown in Table 1, the normalized energy consumption with cycloid trajectory is defined as 100%. Our optimization trajectory performs best in terms of energy consumption.

#### 4.2 Period & Duty Comparison

This experiment is to verify the benefit of Period & Duty optimization. The robot walks on ground as Figure 12 shows, first using a fixed period & duty, and again using the optimized period & duty of Section 3.2. results are shown in Figure 13.

We designed a movement process containing varying velocities. The control group used fixed gait parameters (Figure 13 (a1) for the velocity, Figure 13(b1) for period & duty, and Figure 13(c1) for gait sequence. Period=0.7 s, Duty=0.52). The experimental group used the optimized gait period & duty based on the robot state (Figure 13(a2) for the velocity, Figure 13(b2) for period & duty, and Figure 13(c2) for gait sequence). For

**Table 1** Normalized energy consumption of different swing trajectories

Swing trajectory	Cycloid	Symmetric polynomial	Optimize trajectory
Energy consumption (%)	100	86.33	81.96

simplification, we did not consider turning in our optimization model, so in this experiment we converted the equivalent step size of turning to the translation velocity. The experiment proves that at the given speed, the average joint power (Figure 13(d)) of the optimized gait period & duty is smaller, reduced by 15.9%.

#### 4.3 Gait Adaptor for Stroll

This experiment is to verify the feasibility of the Gait Adaptor for online gait sequence optimization. We carried out a strolling experiment with the Gait Adaptor proposed in Section 3.3, that all step cycles are launched automatically. The motion commands are randomly generated by a handle remoter. The gait patterns at different stages are shown in Figure 14. This experiment proves that our Smart Gait algorithm can generate adaptive gait

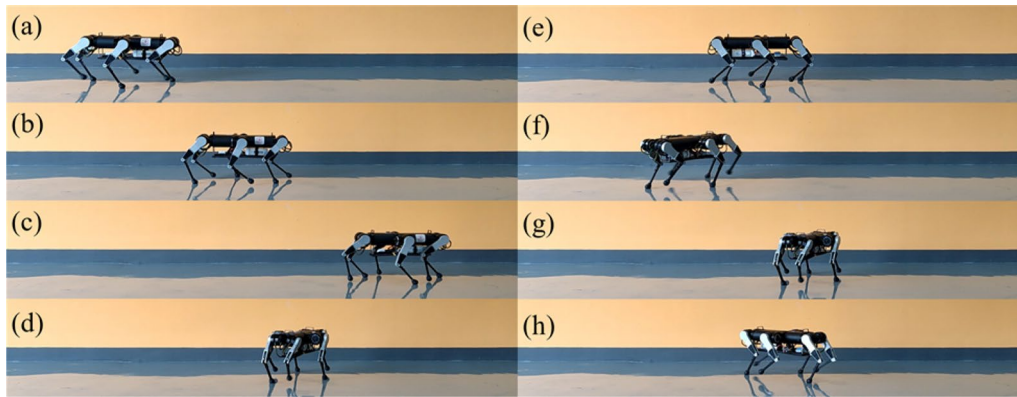


Figure 12 Screenshot of the experiment for period & duty comparison

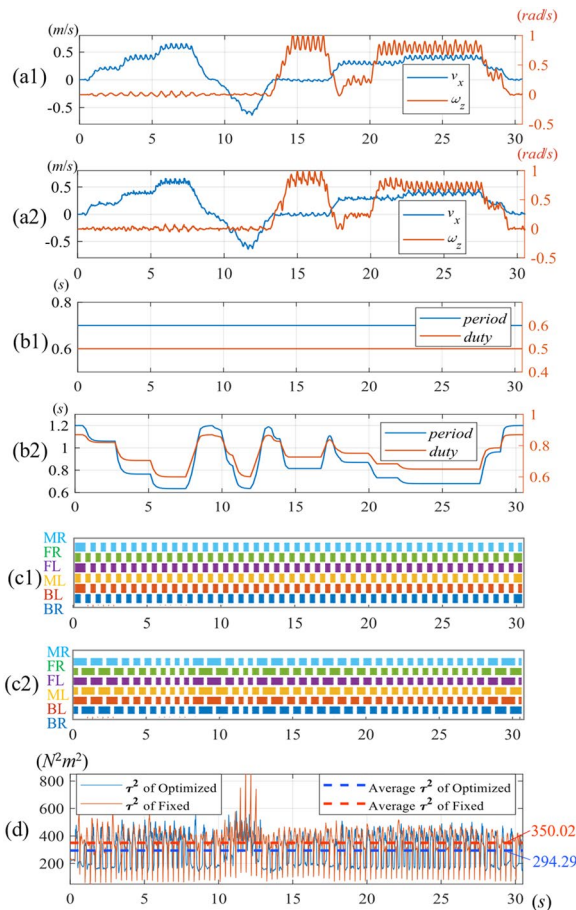


Figure 13 Results of optimized period & duty experiment: (a1), (a2) The estimated velocity and angular velocity of the robot, (b1), (b2) Period & duty, (c1), (c2) Gait cycle, (d) The real time and the average torque squared

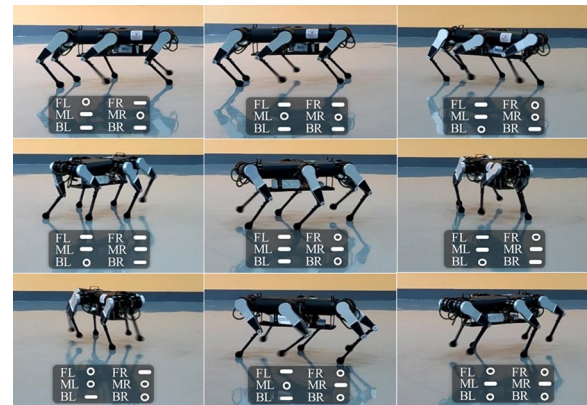
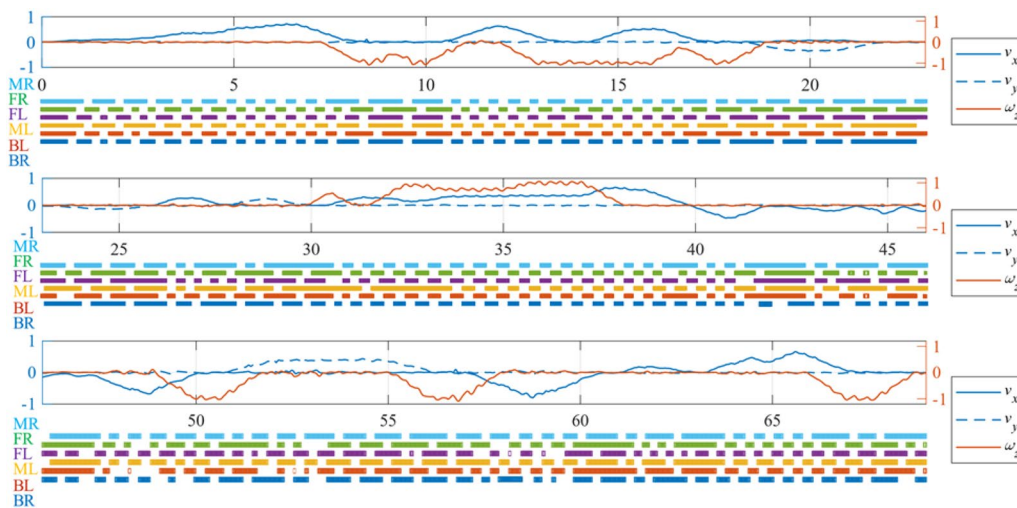


Figure 14 Screenshots of different gait stage from the stroll (Circles for swing phase and rods for contact phase)

sequences of 3D motions. As can be seen in Figure 15, the gait of the robot is not in a fixed or preset pattern, but totally determined by the current state of the robot and contains a variety of gait transitions, which shows out vivid characteristics as living myriapods.

#### 4.4 Gait Adaptor for Disturbance Recovery

This experiment is to verify the anti-interference of the Gait Adaptor for online gait sequence optimization. When challenged by an external interference, the robot needs to take an adaptive gait to maintain its balance. The easiest way is to set a state switch by an < if...else... > judgement, but this human designed script is just a technical method rather than driven by the inherent essence. Here by hiring our Smart Gait framework, the robot successfully makes instinctive responses to two challenges of external interference.



**Figure 15** Estimated velocity of the robot and Smart Gait cycle for the strolling

In the first challenge, as shown in Figure 16, when the front-right leg of the robot is dragged out, the Smart Gait will make the robot automatically lift the leg back to the original position. And as the balance of the robot body is also affected by the dragging, the robot voluntarily lifts the diagonal back-left leg to maintain its symmetrical balance.

In the second challenge, as shown in Figure 17, the robot was impacted by a side kick when standing statically, and the Smart Gait quickly makes an adaptive leg raising motion response. Figure 17(e) shows the gait sequence of the robot, which shows the Smart Gait can handle unexpected interferences online based on the robot’s real-time state.

**4.5 Gait Adaptor for Trench Crossing**

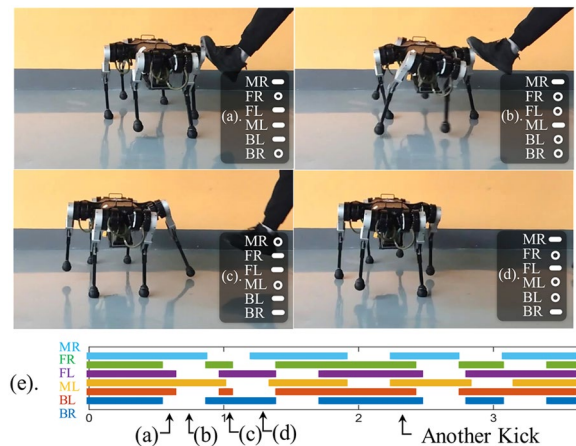
This challenge is to verify the uneven terrain adaptability of the Gait Adaptor for online gait sequence optimization. As shown in Figure 18 and Figure 19, the gap between these two boards is 0.25 m’s width, with a virtual safe margin of 0.05 m for both sides. So, the virtual trench is 0.35 m’s width, a little wider than half of the robot’s length. In this experiment, the position of the trench is pre-known by the robot. The robot moves forward in

0.4 m/s to cross the trench (Figure 18), and again, moves back off in 0.5 m/s (Figure 19).

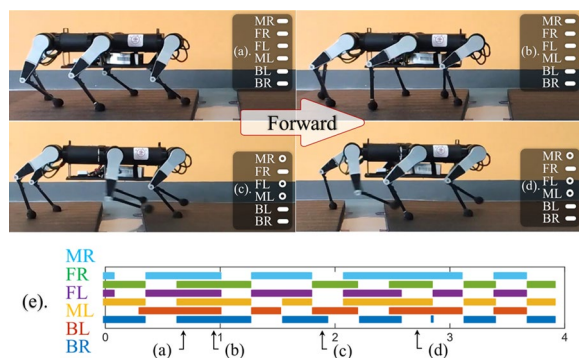
When the robot approaches the trench, it stops at its feet and leans forward to accumulate strength. Then it uses two front legs to stride across the trench, after that two middle legs lift and swing in the same phase, and the back legs transition to staggered phases to stride over again. The gait cycle is shown in Figure 18, in which the coherent movements of the robot are very similar to the cross-trench actions as an organism does. Even more, as Figure 19 shows, when using a faster speed moving back, the robot performs a quasi-jump to overcome the limitation of the leg working space, as the living animals often does. These actions are totally generated by the Smart Gait without any human modification (Additional file 6).



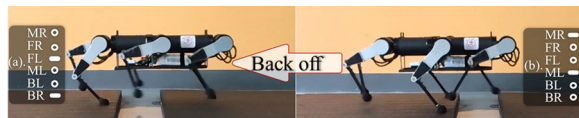
**Figure 16** Disturbance recovery from foot drags



**Figure 17** Disturbance recovery from body kicks: (a)–(d) Screenshots of different gait stages, (e) Smart Gait cycle



**Figure 18** Moving forward to cross a trench (0.4 m/s): (a) Approaching, (b) Holding and gearing up, (c) Striding with middle legs, (d) Striding with back leg, (e) Smart Gait cycle



**Figure 19** Moving back off to cross a trench (0.5 m/s): (a) Striding with middle legs, (b) quasi jump with front legs, (c) Smart Gait cycle

## 5 Conclusions

- (1) This paper has presented the Smart Gait framework, a meaningful work for gait optimization of hexapod robots. The Smart Gait has three modules: the swing trajectory optimization, the period & duty optimization, and the online Gait Adapted for gait transition.
- (2) Smart Gait enables the hexapod robot *LittleStrong* to save more energy during locomotion. Promising results are shown in the experiments. Swing trajectory optimization reduces energy consumption of the swing leg to 81.96%. Period & duty optimization saves 15.9% of the energy consumption of the robot.
- (3) Gait sequence optimization helps the robot to perform smart behaviors like living-creatures, rather than repeating stubborn steps. It helps to generate adaptive gait patterns both in regular and challenge environments in 3D movements for the first time. It also can generate adaptive actions for the hexapod when facing external interferences.
- (4) In the future, we will introduce environmental awareness to our Smart Gait framework, enabling the robot to adapt to more complex terrains, and further improve the bionic behavior intelligence of legged robots.

## Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s10033-024-01000-0>.

**Additional file 1.** Joint velocities and torques of the Back-Right leg of the robot under different swing trajectories for Experiment 4.1.

**Additional file 2.** Results of optimized period & duty for Experiment 4.2.

**Additional file 3.** Estimated velocity of the robot and Smart Gait cycle for the strolling for Experiment 4.3.

**Additional file 4.** Gait stages of the disturbance recovery with body kicks for Experiment 4.4.

**Additional file 5.** Gait stages of moving forward and back off to cross a trench for Experiment 4.5.

**Additional file 6.** Video of the introduction and experiments of the Smart Gait.

## Acknowledgements

Not applicable.

## Authors' Contributions

FG was in charge of the whole trial; YY wrote the manuscript; YX, YZ and QS assisted with sampling and laboratory analyses. All authors read and approved the final manuscript.

## Funding

Supported by National Key Research and Development Program of China (Grant No. 2021YFF0306202).

## Availability of Data and Materials

All data generated or analysed during this study are included in this published article.

## Declarations

### Ethics Approval and Consent to Participate

Not applicable.

### Consent for Publication

Not applicable.

### Competing Interests

The authors declare no competing financial interests.

Received: 5 July 2023 Revised: 16 October 2023 Accepted: 19 January 2024

Published online: 27 February 2024

## References

- [1] S C Niquille. Regarding the pain of SpotMini: Or what a robot's struggle to learn reveals about the built environment. *Architectural Design*, 2019, 89(1): 84-91.
- [2] M Hutter, C Gehring, D Jud, et al. ANYmal - A highly mobile and dynamic quadrupedal robot. *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [3] T Miki, J Lee, J Hwangbo, et al. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 2022, 7(62): eabk2822.
- [4] C Semini, N G Tsagarakis, E Guglielmino, et al. Design of HyQ - a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 2011.
- [5] S Seok, A Wang, Y C Meng, et al. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot. *IEEE*, 2013.

- [6] H W Park, P M Wensing, S Kim. High-speed bounding with the MIT Cheetah 2: Control design and experiments. *International Journal of Robotics Research*, 2017, 36(2): 027836491769424.
- [7] G Bledt, M J Powell, B Katz, et al. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot. *IEEE International Conference of Intelligent Robots (IROS) 2018*, 2018.
- [8] B Katz, J D Carlo, S Kim. Mini Cheetah: A platform for pushing the limits of dynamic quadruped control. *2019 International Conference on Robotics and Automation (ICRA)*, 2019.
- [9] P Yang, F Gao. Leg kinematic analysis and prototype experiments of walking-operating multifunctional hexapod robot. *ARCHIVE Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996 (vols 203-210)*, 2014, 228(12): 2217-2232.
- [10] Y Zhao, F Gao, Q Sun, et al. Terrain classification and adaptive locomotion for a hexapod robot Qingzhui. *Frontiers of Mechanical Engineering*, 2021, 16: 271-284.
- [11] L Mao, F Gao, Y Tian, et al. Novel method for preventing shin-collisions in six-legged robots by utilising a robot-terrain interference model. *Mechanism and Machine Theory*, 2020, 151: 103897.
- [12] C Gehring, S Coros, M Hutler, et al. Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics & Automation Magazine*, 2016, 23(1): 34-43.
- [13] G Bhardwaj, U Mishra, N Sukavanam, et al. Planning adaptive brachistochrone and circular arc hip trajectory for a toe-foot bipedal robot going downstairs. *Journal of Physics: Conference Series*, 2021, 1831(1): 012032.
- [14] D J Hyun, S Seok, J Lee, et al. High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the MIT Cheetah. *International Journal of Robotics Research*, 2014, 33(11): 1417-1445.
- [15] X Zeng, S Zhang, H Zhang, et al. Leg trajectory planning for quadruped robots with high-speed trot gait. *Applied Sciences*, 2019, 9(7): 1508.
- [16] T Liao, S Ye, L Chen, et al. Energy efficient swing leg trajectory planning for quadruped robots walking on rough terrain. *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2019: 2128-2133.
- [17] R Khusainov, A Klimchik, E Magid. Swing leg trajectory optimization for a humanoid robot locomotion. *International Conference on Informatics in Control, Automation and Robotics*, 2016, 3: 130-141.
- [18] X B Chen, F Gao. Energy expenditure of trotting gait under different gait parameters. *Chinese Journal of Mechanical Engineering*, 2017, 30(4): 943-950.
- [19] Z Chen, F Gao, Q Sun, et al. Ball-on-plate motion planning for six-parallel-legged robots walking on irregular terrains using pure haptic information. *Mechanism and Machine Theory*, 2019, 141: 136-150.
- [20] Z Chen, Q Xi, F Gao, et al. Fault-tolerant gait design for quadruped robots with one locked leg using the GF set theory. *Mechanism and Machine Theory*, 2022, 178: 105069.
- [21] Y Fukuoka, Y Habu, T Fukui. A simple rule for quadrupedal gait generation determined by leg loading feedback: A modeling study. *Scientific Reports*, 2015, 5(1): 1-11.
- [22] A J Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 2008, 21(4): 642-653.
- [23] W Ouyang, H Chi, J Pang, et al. Adaptive locomotion control of a hexapod robot via bio-inspired learning. *Frontiers in Neurobotics*, 2021, 15: 627157.
- [24] C Boussema, M J Powell, G Bledt, et al. Online gait transitions and disturbance recovery for legged robots via the feasible impulse set. *IEEE Robotics & Automation Letters*, 2019, 4(2): 1611-1618.
- [25] P Fankhauser, C D Bellicoso, C Gehring, et al. Free gait—An architecture for the versatile control of legged robots. *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016: 1052-1058.
- [26] A W Winkler, C D Bellicoso, M Hutter, et al. Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robotics and Automation Letters*, 2018, 3(3): 1560-1567.
- [27] D Valouch, J Faigl. Gait-free planning for hexapod walking robot. *2021 European Conference on Mobile Robots (ECMR)*, 31 Aug.-3 Sept. 2021, 2021: 1-8, doi: <https://doi.org/10.1109/ECMR50962.2021.9568834>.
- [28] Y P Yin, Y G Xiao, F Gao. Footholds optimization for legged robots walking on complex terrain. *Frontiers of Mechanical Engineering* 2023, 18(2): 26. <https://doi.org/10.1007/s11465-022-0742-y>.
- [29] P Gerum. Xenomai-Implementing a RTOS emulation framework on GNU/Linux. *White Paper, Xenomai*, 2004: 81.
- [30] P Hintjens. *ZeroMQ: messaging for many applications*. O'Reilly Media, Inc., 2013.
- [31] C D Bellicoso, F Jenelten, P Fankhauser, et al. Dynamic locomotion and whole-body control for quadrupedal robots. *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2017.
- [32] A Ranganathan. The levenberg-marquardt algorithm. *Tutorial on LM Algorithm*, 2004, 11(1): 101-110.
- [33] Q Sun, F Gao, X B Chen. Towards dynamic alternating tripod trotting of a pony-sized hexapod robot for disaster rescuing based on multi-modal impedance control. *Robotica*, 2018, 6(7): 1048-1076.
- [34] M Focchi, A Del Prete, I Havoutis, et al. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots*, 2017, 41(1): 259-272.

**Yunpeng Yin** Born in 1995, is currently a PhD candidate at *State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, China*. He received his bachelor degree from *Xi'an Jiaotong University, China*, in 2018. His research interests include legged robots and automation control.

**Feng Gao** Born in 1956, is currently a professor at *Shanghai Jiao Tong University, China*. His research interests include mechanism, machine theory and legged robots.

**Qiao Sun** Born in 1989, is currently an employee of *Lenovo Corporation, China*. He received his doctor degree from *Shanghai Jiao Tong University, China*, in 2018.

**Yue Zhao** Born in 1992, is currently a postdoc at *AI Institute, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China*. He received his doctor degree from *Shanghai Jiao Tong University, China*, in 2021.

**Yuguang Xiao** Born in 1992, is currently a PhD candidate at *State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, China*.