



# Obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III

**Yue Zhao, Xun Chai, Feng Gao\*, Chenkun Qi**

State Key Laboratory of Mechanical System and Vibration, Department of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China



## HIGHLIGHTS

- A novel obstacle avoidance path planning method for a hexapod robot is proposed.
- Formulate the obstacle avoidance issue as an optimization problem.
- Consider the walking stability and the kinematic feasibility simultaneously.
- The proposed scheme is applied, and several experiments are carried out.

## ARTICLE INFO

### Article history:

Received 17 May 2017

Received in revised form 5 December 2017

Accepted 17 January 2018

Available online 12 February 2018

### Keywords:

Obstacle avoidance

Hexapod robot

Trajectory planning

Gait motion generation

Parallel mechanism

## ABSTRACT

Legged robots have advanced potential to move in complex environment accomplishing operating, rescuing and detecting tasks. In real applications, bypassing large obstacles is a more common choice for legged robots comparing with walking over and climbing the obstacles. However, few papers involve the obstacle avoidance approach for legged robots. An obstacle avoidance and motion planning scheme for a hexapod robot is presented in this paper. The scheme takes advantage of the superior mobility of the legged robot and fulfills requirements of walking stability and kinematic feasibility. Firstly, a novel obstacle avoidance trajectory planning method is proposed, which is inspired by the superior mobility of the legged robot. Then, a motion generation approach for the legged robot is developed to control the robot to walk along the planned trajectory. The approach coordinates the body motion and the feet motions to fulfill requirements of walking stability and kinematic feasibility simultaneously. Finally, the scheme is integrated on a hexapod robot and tested by real experiments.

© 2018 Elsevier B.V. All rights reserved.

## 1. Introduction

Frequent occurrence of natural and man-made disasters greatly prompts public desire for the robots' help. For operating, detecting and rescuing applications, mobile robots are so valuable that they could take the place of human beings to enter into the hazardous environment. In such applications, legged robots have advanced potential to navigate in complex environment compared with their wheeled counterparts [1]. Because they only need some discrete footholds for walking. In recent years, researches have attached more and more importance on legged robots [2–5]. While executing tasks in complex environment, legged robots can choose to bypass, walk over or climb the obstacles based on different features of the obstacles. Most related works [6–12] concentrate on control schemes which enable legged robots to walk over obstacles and

traverse slightly uneven terrain. However, bypassing large obstacles is a more common choice for legged robots in real application environment, few papers involve the obstacle avoidance approach for legged robots.

In the issue of robotic autonomous navigation, several typical obstacle avoidance methods have been proposed in the past decades. Khatib [13] originally introduced the artificial potential field (APF) method. In the APF method, the obstacles produce a repulsive force, while the target produces an attractive force, so that the robot can reach the target without colliding with any obstacles. Lavalle [14] introduced the concept of the Rapidly-exploring Random tree (RRT) as a randomized data structure. RRT is specially designed to handle the nonholonomic constraints and high degrees of freedom. Borenstein and Koren [15] developed the virtual force field (VFF) method, which permitted the detection of unknown obstacles simultaneously with the steering of the mobile robot to avoid collisions and advancing towards the target. The VFF method integrates two important concepts: certainty grids for obstacle representation and potential field for navigation. Johann Borenstein et al. [16,17] proposed the classical VFH and VFH+

\* Corresponding author.

E-mail addresses: [yue.zhao@sjtu.edu.cn](mailto:yue.zhao@sjtu.edu.cn) (Y. Zhao), [xunchai@outlook.com](mailto:xunchai@outlook.com) (X. Chai), [fengg@sjtu.edu.cn](mailto:fengg@sjtu.edu.cn) (F. Gao), [chenkqi@sjtu.edu.cn](mailto:chenkqi@sjtu.edu.cn) (C. Qi).

obstacle avoidance method for the wheeled robot. The VFH method uses 2D Cartesian histogram grid to represent the real world, and applies a two-stage data reduction process to solve the obstacle avoidance problem. In the first stage, the histogram grid is reduced to a one-dimensional polar histogram, which stores a value representing the obstacle density. In the second stage, the algorithm selects the most suitable sector with a lower obstacle density. The VFH+ method offers several improvements that result in smoother robot trajectories and greater reliability.

In the related application field, obstacle avoidance has been extensively integrated into the wheeled robots. Sezer and Gokasan [18] designed an obstacle avoidance algorithm named “follow the gap method”, which was easy to tune and taken into consideration the field of view and the nonholonomic constrains of the wheeled robots. Hoeller et al. [19] proposed a local navigation planning approach for collision avoidance, which used motion prediction and probabilistic roadmaps to plan collision-free paths to a given target location relative to the robot. Rashid et al. [20] introduced a multi-robot collision free navigation method based on reciprocal orientation, which generated both smooth and collision-free robot trajectories and can be applied in dynamic environments. Cherubini and Chaumette [21] proposed and validated a framework for visual navigation with collision avoidance for a wheeled mobile robot. Their control scheme guaranteed that the navigation and the obstacle avoidance can be achieved simultaneously. Yu Lwin and Yamamoto [22] designed an obstacle-responsive navigation scheme for a wheeled mobile robot based on look-ahead control. The look-ahead control method fused the goal-aiming and the collision avoiding task together by changing coordinate frames and associated elements. Chun-Hsu Ko et al. [23] developed a scheme based on parametric trajectory planning. Their method considered the collision avoidance as the constraint, and then a feasible collision-free trajectory was derived by solving an unconstrained optimization problem. Zhu et al. [24] proposed a distance histogram bug algorithm for mobile robots in unknown environments containing moving obstacles. Their method improved the traditional bug algorithm by considering some previous limitations. Hongjiu Yang et al. [25] generated a prudential function to solve the problems of obstacle avoidance for a wheeled mobile robot and a nonlinear controller is designed. Martin Wermelinger et al. [26] proposed a cascaded planning structure makes use of different levels of simplification to allow for fast search in simple environments, while retaining the ability to find complex solutions, such as paths through narrow passages. Thi Thoa Mac et al. [27] developed a simple algorithm for robot obstacle avoidance using bumper event. Kai-Tai Song et al. [28] proposed a shared-control scheme combines active obstacle avoidance and passive-compliant motion commands.

Recently, Tai Lei and Liu Ming [29,30] have proposed a deep-network solution towards the obstacle avoidance of the mobile robot. They built a complex hierarchical deep-network, which comprises a CNN front-end network for perception and a fully-connected network for decision making. The input of the network is the raw depth sensor data, and the output is the robot control commands. The contribution of their work is that the similar deep-network method can be used to solve related model-less robotic problems. The deep-network method needs to build complex hierarchical network model, and use a lot of experimental data to train the model. The training process is complicated, and maybe need some cumbersome operations under the human beings' supervisions. In this paper, we mainly focus on proposing a direct, simple and effective method to solve the obstacle avoidance problem of a complex legged robot system.

Hexapod Walking Robot(HWR) may have two basic body shapes: rectangular and hexagonal [31]. The first one has six legs distributed symmetrically along two sides, each side having three

legs. The second has legs distributed axi-symmetrically around the body, in a hexagonal or circular shape. The tripod gait has been the first successful gait. In 1972, a computer-controlled walking machine with electric drives was constructed at University of Rome [32]. The free gait is suitable for terrains, and a non-periodic gait, called free gait was introduced in [33]. COMET hexapods are designed to operate on untrusted terrain [34]. LAURON hexapod's control architecture was based on neural networks [35]. Giuseppe Carbone et al. [36] proposed a preliminary study on the application of a hybrid multi-objective optimization approach which identifies minimizing the energy consumption and efficiency as well as maximizing the walking speed and the size of obstacles that a leg can overtake as its criteria. However, the robot discussed in this paper belongs to bilateral symmetry but not rectangular. The differences of walking ability in different directions is accounted in path planning.

To the best of the authors' knowledge, few papers involve the obstacle avoidance for legged robots. Several key characteristics of legged robots make the obstacle avoidance methodology typically different from that of wheeled robots. Wheeled robots have non-holonomic mobility, they cannot move along arbitrary direction. For example wheeled robots cannot move laterally to avoid an obstacle. While legged robots have the 6-dimension motion ability, theoretically they are able to walk towards any direction. Compared with wheeled counterparts, legged robots are more agile and have the “entire” mobility, which originally makes them have a better obstacle avoidance ability. However, on the other hand, the motion agility also means the motion planning for legged robots is more complicated. What is more, the geometry shape of the obstacle in reality is very irregular, and the legged robot itself generally has a complex body shape. So seeking an appropriate walking direction based on the relationship between the legged robot and the obstacle is difficult. Walking stability and kinematic feasibility are extremely important for legged robots, which makes the motion planning more complex. The obstacle avoidance, the motion planning with the constraints of walking stability and kinematic feasibility are needed to be considered at the same time. Therefore, the obstacle avoidance methodology for legged robots remains a challenging problem. To supplement relevant study in legged robots, we propose a novel obstacle avoidance and motion planning scheme for a hexapod robot. This paper makes the following contributions:

1. A novel obstacle avoidance path planning method for a hexapod robot Octopus-III is proposed. The method takes advantage of the superior mobility of legged robots. It is simple and flexible, which computes the walking direction by formulating the obstacle avoidance issue as an optimization problem.

2. The motion generation approach is developed to control the legged robot to walk along the planned trajectory. The approach satisfies the requirements of obstacle avoidance, walking stability and kinematic feasibility at the same time.

3. We integrate the proposed scheme on the hexapod robot Octopus-III. Real experiments are carried out to validate the methodology.

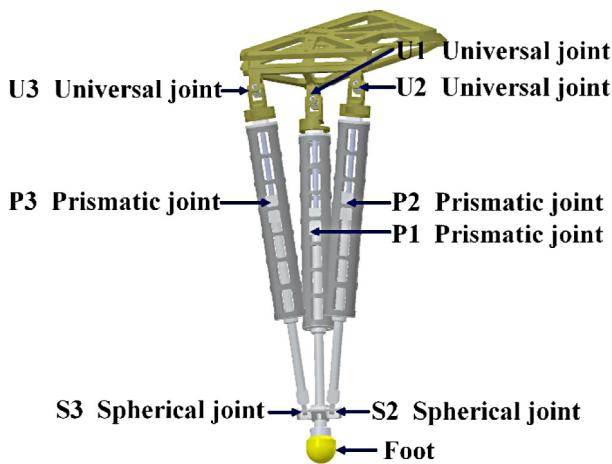
The remainder of this paper is organized as follows. Section 2 provides a brief introduction of the robot system. Section 3 presents the obstacle avoidance trajectory planning method in detail. The motion generation approach is developed in Section 4. After, the proposed scheme is validated by real experiments in Section 5. Finally, Section 6 concludes the paper.

## 2. Robot system model

This paper proposes an obstacle avoidance and motion planning scheme for the hexapod robot Octopus-III. Octopus-III is a prototypic, six-legged walking robot which is intended to accomplish



**Fig. 1.** The Octopus-III robot.



**Fig. 2.** The mechanism design of the leg.

rescuing and detecting tasks in harsh environment. Walking safely without colliding with any obstacles is extremely prerequisite for Octopus-III in such environment.

As Fig. 1 shows, Octopus-III has a hexagon body with six similar legs symmetrically distributed around the body. In the view of mechanism point, Octopus-III is a six DOFs (degrees of freedom) parallel-parallel moving platform integrated walking and manipulating. It has a total mass of 240 kg, it is about 1m high and its feet span an area of 1.3 m × 0.9 m.

Fig. 2 shows the detailed mechanism design of the robot leg. Its leg is a parallel mechanism having three chains. As Fig. 2 shows, the  $U_2P_2S_2$  chain and the  $U_3P_3S_3$  chain have the same mechanical structure and dimension size. Each of them is constructed by a universal joint, a prismatic joint and a spherical joint. Another chain  $U_1P_1$  is constructed by a universal joint and a prismatic joint. Each leg has three active prismatic joints. The liner movement of the prismatic joint is achieved by the ball-bearing screw, which is actuated by the motor. The special parallel-parallel mechanism design concept provides Octopus-III with outstanding walking stability and load capacity.

As Fig. 3 shows, Octopus-III hosts a variety of proprioceptive sensors including a RGB-D camera, an IMU, a compass, a GPS, and a six-axis force/torque sensor. The core computer of the whole control system is an industrial PC running the real-time Linux OS. The RGB-D camera, which is connected to the core computer by USB, detects the terrain in front of the robot. The six-axis force/torque sensor connected to the core computer by Ethernet

measures all six components of force and torque of the body. The GPS and the compass help the robot locate itself and navigate the right direction in outdoor environment. The IMU measures the inclination, the angle velocity and the linear acceleration of the robot. The core computer sends planned data to drivers via EtherCAT. Drivers provide current to motors, and servo control motors using the feedback data from resolvers.

The motion control circle of Octopus-III is 1 ms and the walking accuracy is within 1 cm. The obstacle detection cycle is about 30 ms and the accuracy is about 1 cm. This robot can climb stairs with 37.5 cm long an 16.0 cm high step which is about 23°. A 20.0 cm high step can be strode and a 60.0 cm long ditch can be stepped over. Almost 500.0 kg load which is about 2 times its weight can be supported. This robot's targets are walking at 0.3 m/s and stepping over 20° stair, which is a little better compared with other weighty hexapod robots, like COMET-IV [37]. And the results of these experiments indicate the targets are achieved.

### 3. Obstacle avoidance for the hexapod robot

#### 3.1. The geometric modeling

Geometric models of the robot and the obstacle are indispensable for calculating the distance between them. The selection of which type of geometric models to use is related to the specific obstacle avoidance problem. This paper primarily focuses on the distance between the robot and the obstacle. Therefore, the circle-based geometric model is selected in our method because of its efficiency and simplicity.

As Fig. 1 shows, Octopus-III has a hexagonal body, two circles are used to envelope its body. As Fig. 4 shows, the front part of the robot is enveloped by a circle with the center of  $(x_{r1}, y_{r1})$  and the radius of  $r_{r1}$ . The back part of the robot is enveloped with a circle whose center is  $(x_{r2}, y_{r2})$  and radius is  $r_{r2}$ . Whenever the robot moves, the positions of the two circles are updated with new values.

Similar to the robot, the obstacle is also represented by the circle model. Assuming that the obstacle is represented by a circle with the center of  $(x_{obs}, y_{obs})$  and the radius of  $r_{obs}$ . As Fig. 5 shows, the distance to the obstacle will be used to generate an obstacle avoidance trajectory in the following section. For this reason, the formal definition of the distance between the robot and the obstacle is defined by the following equation:

$$d_{pq} = \sqrt{(x_{rp} - x_{obsq})^2 + (y_{rp} - y_{obsq})^2} - r_{rp} - r_{obsq} \quad (1)$$

where  $d_{pq}$  is the distance between the pth circle model of the robot and the qth obstacle model.  $(x_{rp}, y_{rp})$  is the position of the pth circle model of the robot.  $(x_{obsq}, y_{obsq})$  is the position of the qth obstacle model.  $r_{rp}$  and  $r_{obsq}$  are the radii of the pth circle model of the robot and the qth obstacle model respectively.

#### 3.2. Obstacle avoidance path planning

The obstacle avoidance problem we need to solve is as follows: the start position and the target position are given, computing the robot's moving trajectory so that it can walk to the target position without colliding with any obstacles.

The proposed obstacle avoidance method takes advantage of the excellent mobility of Octopus-III. Octopus-III is able to walk towards any directions. Fig. 6 shows the walking capability of Octopus-III along different directions. Vectors  $w_0 \dots w_n$  represent the walking capacity in different directions. The length of the vector denotes the walking speed. The obstacle avoidance method seeks a safe and feasible trajectory to the target position without colliding with any obstacles by making the robot walk along a series of pre-defined directions.

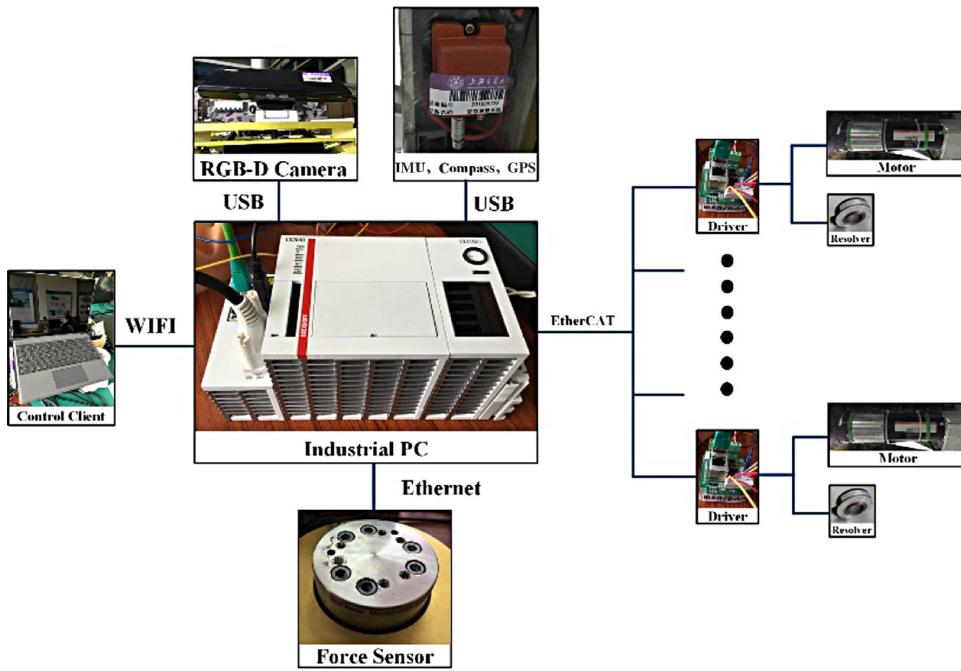


Fig. 3. Sensor and computer system.

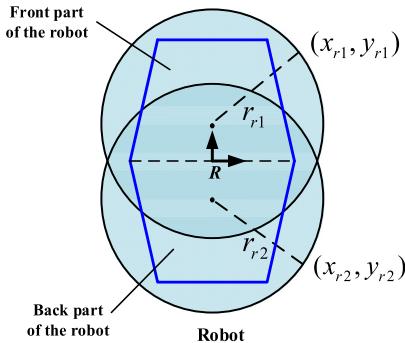


Fig. 4. Geometric model of the robot.

**Fig. 7** basically illustrates the obstacle avoidance method. The start position and the target position are known. The position and the size of the obstacle are detected based on other algorithms, which beyond the range of this article and will not be presented here. As **Fig. 7** shows, the whole walking process contains several sub-processes. In every sub-process, the robot walks towards a pre-defined direction. The optimization method is used to search the most appropriate walking direction which balances the motion towards the target position and the distance between the robot and obstacles. Then the selected walking direction is applied to the robot for next several time-steps. And the method is repeated until the robot reaches the target position.

As **Fig. 6** shows, the pre-defined walking directions of the robot are  $\mathbf{w}_0 \dots \mathbf{w}_n$ , whose unit vectors are  $\mathbf{u}_0 \dots \mathbf{u}_n$  respectively. Besides above walking directions, the unit vector  $\mathbf{u}_T$  is introduced to represent the direction towards the target position.  $\mathbf{u}_T$  is obtained from the following equation:

$$\mathbf{u}_T = \frac{(x_T, y_T) - (x_R, y_R)}{\sqrt{(x_T - x_R)^2 + (y_T - y_R)^2}} \quad (2)$$

where  $(x_T, y_T)$  is the target position and  $(x_R, y_R)$  is the current position of the robot. The number of pre-defined walking directions

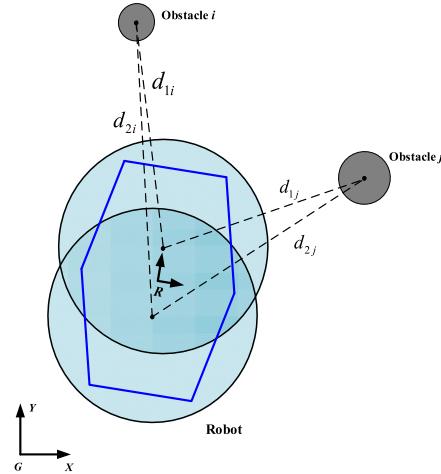


Fig. 5. Calculation of the distance to the obstacle.

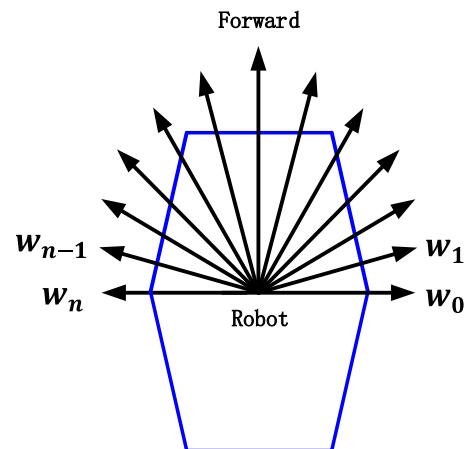
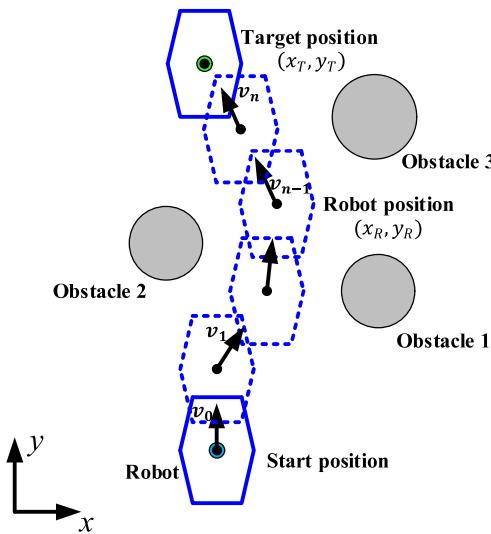


Fig. 6. Walking capability of the robot.



**Fig. 7.** Illustration of the obstacle avoidance method.

$\mathbf{u}_0 \dots \mathbf{u}_n, \mathbf{u}_{n+1}(\mathbf{u}_T)$  are  $n + 1$ . For each selected walking direction  $\mathbf{u}_i$ , the robot will walk towards this direction in the next  $n_T$  time-steps. The walking direction selected issue is formulated as an optimization problem:

$$\min F(\mathbf{u}_i) = \frac{1}{1 + \mathbf{u}_i \cdot \mathbf{u}_T \cdot |\mathbf{w}_i|}, i = 1, \dots, n, n + 1 \quad (3)$$

$$\text{subject to : } d_{\min} < d_{pq}(r). \quad (4)$$

A penalty method is used to replace the constrained optimization problem to a series of unconstrained problems whose solutions ideally converge to the solution of the original constrained problem. To describe how safety the distance is, a penalty parameter  $c_{pq}(r)$  is defined.

$$c_{pq}(r) = \begin{cases} 1 & d_{pq}(r) \leq d_{\min} \\ \frac{1}{2} \left[ 1 + \tanh \left( \frac{1}{d_{pq}(r) - d_{\min}} \right) \right] & d_{\min} < d_{pq}(r) < d_{\max} \\ 0 & d_{pq}(r) \geq d_{\max} \end{cases} \quad (5)$$

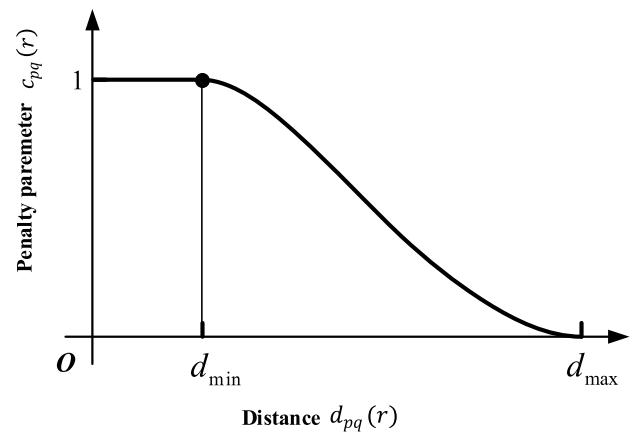
where  $d_{pq}(r)$  is the distance between the  $p$ th circle model of the robot and the  $q$ th obstacle model at the  $r$ th time-step.  $d_{\min}$  is the minimum acceptable distance between the robot and the obstacle. If  $d_{pq}(r)$  is smaller than  $d_{\min}$ ,  $c_{pq}(r)$  is equal to the maximum value 1. It means the distance to the obstacle is very close, the robot should avoid the obstacle.  $d_{\max}$  is the distance limit beyond which the same penalty parameter is used. If  $d_{pq}(r)$  is larger than  $d_{\max}$ ,  $c_{pq}(r)$  is equal to the minimum value 0. It means the distance between the robot to the obstacle is safe enough. **Fig. 8** shows the general graphic representation of the penalty parameter.

The walking direction selected issue is formulated as an optimization problem which contains penalty parameters representing the constraint of the distance between the robot and the obstacle. The object function of each direction is calculated from the following formula:

$$\min F(\mathbf{u}_i) = \frac{1}{1 + \mathbf{u}_i \cdot \mathbf{u}_T \cdot |\mathbf{w}_i|} + \sum_{r=1}^{n_T} \sum_{p=1}^{n_p} \sum_{q=1}^{n_q} c_{pq}(r), \quad (6)$$

$$i = 1, \dots, n, n + 1$$

where  $\mathbf{u}_i$  is the unit vector needs to be selected for the robot to walk.  $n_p$  is the number of circles which envelop the robot, and  $n_q$



**Fig. 8.** Penalty parameter  $c_{pq}(r)$  as a function of the distance  $d_{pq}(r)$ .

is the obstacles' number.  $c_{pq}(r)$  is the penalty parameter used to implement the distance constraint between the  $p$ th circle model of the robot and the  $q$ th obstacle model at the  $r$ th time-step.

The optimization approach to seek the walking direction is heuristic. Concretely, 36 discrete fixed directions evenly spatially distributed ( $0^\circ, 5^\circ, 10^\circ, \dots, 175^\circ, 180^\circ$ ) are defined to act as candidates for the robot to select. Assuming that the robot walks towards a specific direction, distances to obstacles can be obtained and the corresponding object function can be computed. The object function balances the requirement of obstacle avoidance and reaching the target position. The first term in formula (6) describes how much the robot deviates from the target position when it walks towards the selected direction. The minimal value of the first term means that the robot walks almost towards the target position. The second term represents how close the robot is away from the obstacle when it walks towards the selected direction. The closer the robot is to the obstacle, the bigger value of the second term is obtained. The direction  $\mathbf{u}_i$  with the minimum value of  $F(\mathbf{u}_i)$  is selected as the walking direction in the next  $n_T$  time-steps. The method is repeatedly functioned every  $n_T$  time-steps until the robot reaches the target position. The optimization aims to seek the best degree from these 36 directions. Usually, the global optimum can be obtained in 5 ms.

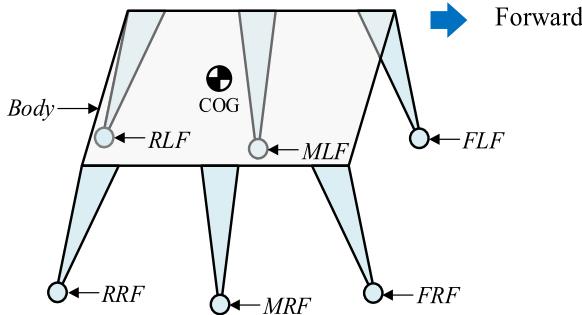
#### 4. Robot motion generation

In this section, we discuss how to control the robot to walk along the planned trajectory. The determination of overmuch inputting DOFs and the requirements of walking stability and kinematic feasibility make the control methodology of legged robots more complicated than that of wheeled robots. Legged robots need to well coordinate the body motion and the feet motions in order to walk with high stability and relatively fast speed. The proposed motion generation method outputs the body trajectory and the feet trajectories, which allows the robot to walk along the planned trajectory while maintaining the walking stability and kinematic feasibility. The method includes three modules, the foot swinging sequence planner, the body motion planner and the foot motion planner, which will be presented in detail in the following sections.

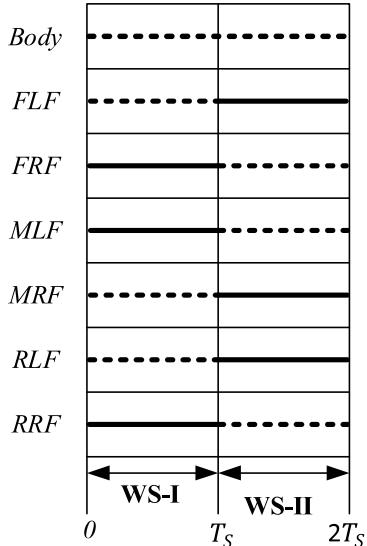
##### 4.1. Foot swinging sequence planner

As mentioned above, the robot is applied to accomplish operating, detecting and rescuing tasks in complex and harsh environment. The robot is intentionally designed to have high load capacity, agile and accurate operating ability. So it is very important for

**FRF:** Front right foot.  
**FLF:** Front left foot.  
**MRF:** Middle right foot.  
**MLF:** Middle left foot.  
**RRF:** Rear right foot.  
**RLF:** Rear left foot.  
**Body:** Robot body  
**COG:** Center of gravity



**Fig. 9.** Definition of different feet.

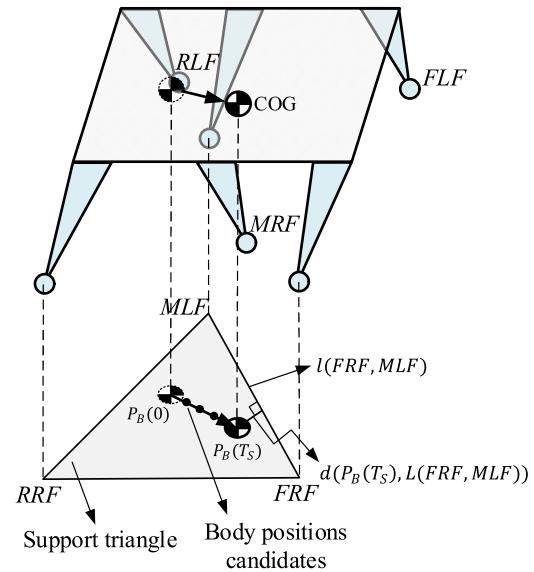


**Fig. 10.** Diagram of gait sequence: foot in support in bold solid line, foot in swinging in bold dashed line.

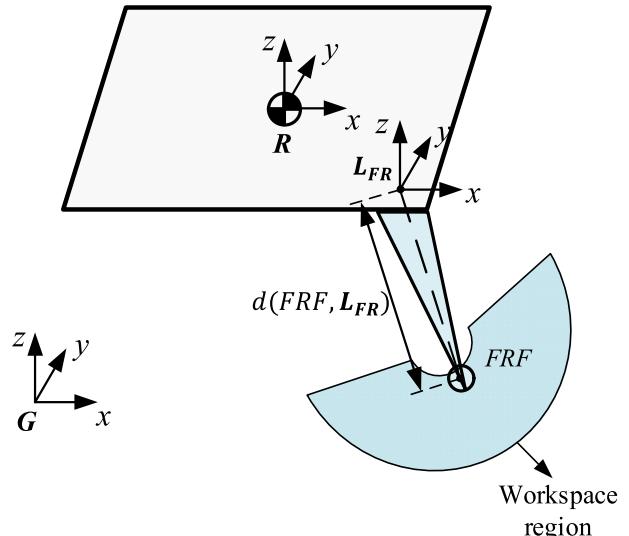
the robot to maintain high body stability all the time. The motion generation method applies the tripod stable gait, which is the fastest stable gait for a hexapod robot. The following nomenclature is used for representing different feet as Fig. 9 shows:

**FRF:** Front right foot.  
**FLF:** Front left foot.  
**MRF:** Middle right foot.  
**MLF:** Middle left foot.  
**RRF:** Rear right foot.  
**RLF:** Rear left foot.  
**Body:** Robot body  
**COG:** Center of gravity

The tripod gait divides six feet into two groups, *FRF*, *MLF*, *RRF* and *FLF*, *MRF*, *RLF*. Each group forms a triangle. During the walking process, each triangle has two states, the support state and the swinging state. Both triangles can be in support state at the same time, but they cannot be in swinging state simultaneously. Fig. 10



**Fig. 11.** Support triangle and stability measurement.



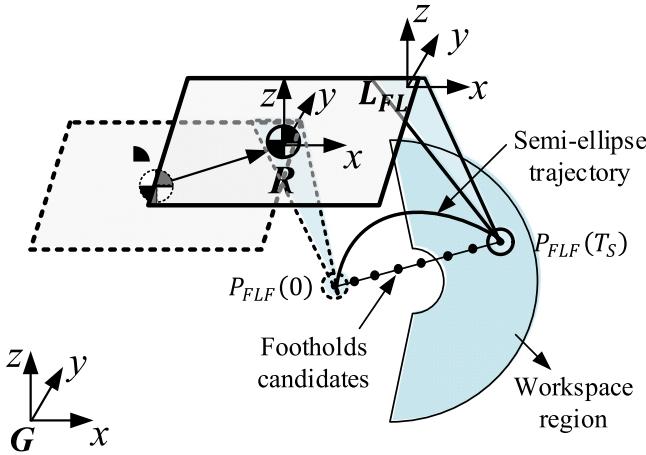
**Fig. 12.** Illustration of the foot workspace.

shows the gait sequence. In order to achieve relatively fast speed, the sequence planner regulates that at any time only one triangle is in support state, while the other triangle is in swinging state.

The complete walking process has two stages as shown in Fig. 10. In the first walking stage (WS-I), *FRF*, *MLF* and *RRF* are fixed on the ground to actuate the body to move along the planned trajectory. At the same time, lifting *FLF*, *MRF*, and *RLF*, moving them to specific positions. In the second walking stage (WS-II), *FLF*, *MRF*, and *RLF* are fixed on the ground continuing to actuate the body to move along the trajectory. At the same time, proceeding to lift *FRF*, *MLF* and *RRF*, move them to specific positions.

#### 4.2. Body motion planner

The cycle the robot spends walking one step is denoted by  $T_S$ . The body motion parameters within  $T_S$  include the walking direction  $\alpha$  and the target position  $P_B(T_S) = (x_B(T_S), y_B(T_S))^T$ .  $\alpha$  is computed based on the approach proposed in Section 3. The



**Fig. 13.** Trajectory planning of the swinging foot.

robot is planned to walk along  $\alpha$  throughout  $T_S$ . In order to achieve relatively fast speed, the robot is regulated to move a long distance within  $T_S$ . The walking stability and the kinematic feasibility are needed to be guaranteed during the walking process.

To guarantee the walking stability, the vertical projection of the COG should be located in the support triangle region constructed by  $MLF$ ,  $RRF$  and  $FRF$  as Fig. 11 shows.  $P_B(0) = (x_B(0), y_B(0))^T$  denotes the initial position of the vertical projection.  $l(FRF, MLF)$

represents the line determined by  $FRF$  and  $MLF$ . Because of the kinematic error of the robot itself, the distance between the vertical projection of COG and the edges of the support triangle is regulated to be larger than a certain value  $\delta_S$ , which is defined as the stability margin.

What is more, the kinematic feasibility is also needed to be considered. The horizontal workspace region of the foot is depicted in Fig. 12.  $G$ ,  $R$ ,  $L_{RF}$  denote the ground coordinate system(G-CS), the robot coordinate system(R-CS) and the right front leg coordinate system( $L_{RF}$ -CS) respectively. The foot workspace is defined as the distance from the foot to the  $L_{RF}$ -CS, which is denoted by  $d(FRF, L_{RF})$ . The position of the  $L_{RF}$ -CS can be calculated from the position of the R-CS. The relationship between the  $L_{RF}$ -CS and the R-CS is  $L_{RF} = {}^{L_R T} \cdot R \cdot {}^{R T}$ .  ${}^{L_R T}$  is the transformation matrix from the R-CS to the  $L_{RF}$ -CS.  ${}^{L_R T}$  is determined by the mechanism size of the robot. Subscripts  $\omega_{min}$  and  $\omega_{max}$  denote the minimum limit and the maximum limit of the workspace.

The body motion planner is formulated to find the farthest distance the robot can move in  $T_S$  within the constraints of the walking stability and the kinematic feasibility. The following optimization problem is used to find the optimum target position:

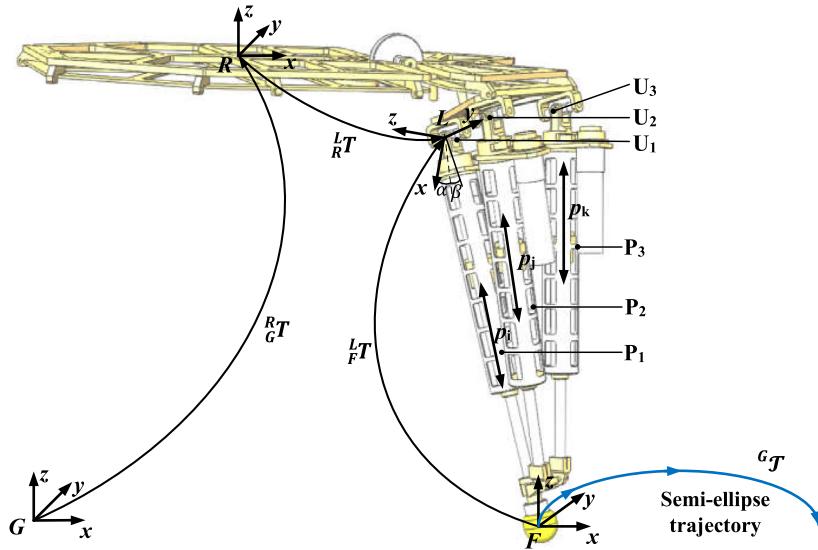
$$\max d(P_B(0), P_B(T_S)) \quad (7)$$

$$\text{subject to : } \frac{y_B(T_S) - y_B(0)}{x_B(T_S) - x_B(0)} = \cot(\alpha) \quad (8)$$

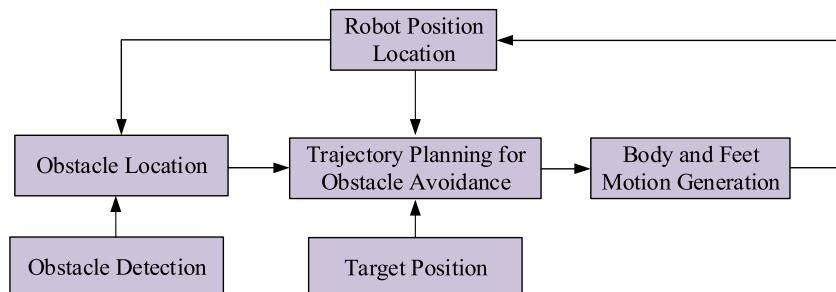
$$d(P_B(T_S), l(FRF, MLF)) > \delta_S \quad (9)$$

$$\omega_{min} \leq d(FRF, L_{RF}) \leq \omega_{max} \quad (10)$$

$$\omega_{min} \leq d(MLF, L_{ML}) \leq \omega_{max} \quad (11)$$



**Fig. 14.** Kinematic computation of the robot.



**Fig. 15.** The control architecture of obstacle avoidance and motion planning.



**Fig. 16.** The first experimental site.

$$\omega_{\min} \leq d(RRF, L_{RR}) \leq \omega_{\max} \quad (12)$$

where  $d(P_B(0), P_B(T_S))$  is the distance the robot moves in  $T_S$ . It is the optimization object. Formula (8) ensures the robot walks along the selected direction  $\alpha$ . Formula (9) guarantees the walking stability, and formulas (10), (11), (12) ensure the kinematic feasibility. To search for the optimum body position, we define a series of finite discrete positions to act as candidates as shown in Fig. 11. The position is valid if it fulfills the constraint conditions. According to the previous analysis, the position with maximum object function value is selected as the optimal.

#### 4.3. Foot motion planner

The foot motion planner consists of two parts, one is the selection of the next foothold, the other one is the trajectory planning of the swinging foot (see Fig. 13).

Similar to the body motion planner, a farther position is selected as the next foothold in order to achieve relatively fast speed while considering the walking stability and the kinematic feasibility. The following optimization problem is formulated to find the optimum footholds:

$$\begin{aligned} & \max[d(P_{FLF}(0), P_{FLF}(T_S)) + d(P_{MRF}(0), P_{MRF}(T_S)) \\ & \quad + d(P_{RLF}(0), P_{RLF}(T_S))] \end{aligned} \quad (13)$$

$$\begin{aligned} \text{subject to : } & \frac{y_{FLF}(T_S) - y_{FLF}(0)}{x_{FLF}(T_S) - x_{FLF}(0)} \\ & = \frac{y_{MRF}(T_S) - y_{MRF}(0)}{x_{MRF}(T_S) - y_{MRF}(0)} \\ & = \frac{y_{RLF}(T_S) - y_{RLF}(0)}{x_{RLF}(T_S) - y_{RLF}(0)} = \cot(\alpha) \end{aligned} \quad (14)$$

$$d(P_B(T_S), l(FLF, MRF)) > \delta_S \quad (15)$$

$$\omega_{\min} \leq d(FLF, L_{FL}) \leq \omega_{\max} \quad (16)$$

$$\omega_{\min} \leq d(MRF, L_{MR}) \leq \omega_{\max} \quad (17)$$

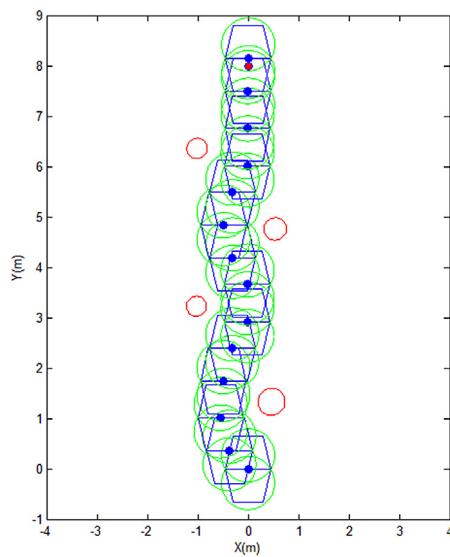
$$\omega_{\min} \leq d(RLF, L_{RL}) \leq \omega_{\max} \quad (18)$$

where  $P_{FLF}(0) = (x_{FLF}(0), y_{FLF}(0))^T$  and  $P_{FLF}(T_S) = (x_{FLF}(T_S), y_{FLF}(T_S))^T$  respectively represent the initial position and the next foothold position of the swinging  $FLF$  in  $T_S$ .  $d(P_{FLF}(0), P_{FLF}(T_S))$  denotes the distance between the initial position and the next foothold position. Formula (14) ensures the stride direction is along the walking direction  $\alpha$ . Formula (15) guarantees the walking stability. Formulas (15), (16), (17) make sure the selected footholds are within the workspace region. The approach to search the optimum foothold is similar to the searching process of body position.

Fig. 14 shows the kinematic computation of the robot. The trajectory of the swinging foot is regulated as a semi-ellipse (denoted by  ${}^6I$ ) in the G-CS.  ${}^L I({}^L x, {}^L y, {}^L z)$  denotes the foot trajectory in the leg coordinate system(L-CS). As mentioned above, each leg has three inputs  $p_i, p_j, p_k$ , which are the lengths of the prismatic joints, needed to be solved in order to actuate the robot to walk along the



**Fig. 17.** Image flows of the first experiment.



**Fig. 18.** The robot trajectory in the first experiment.

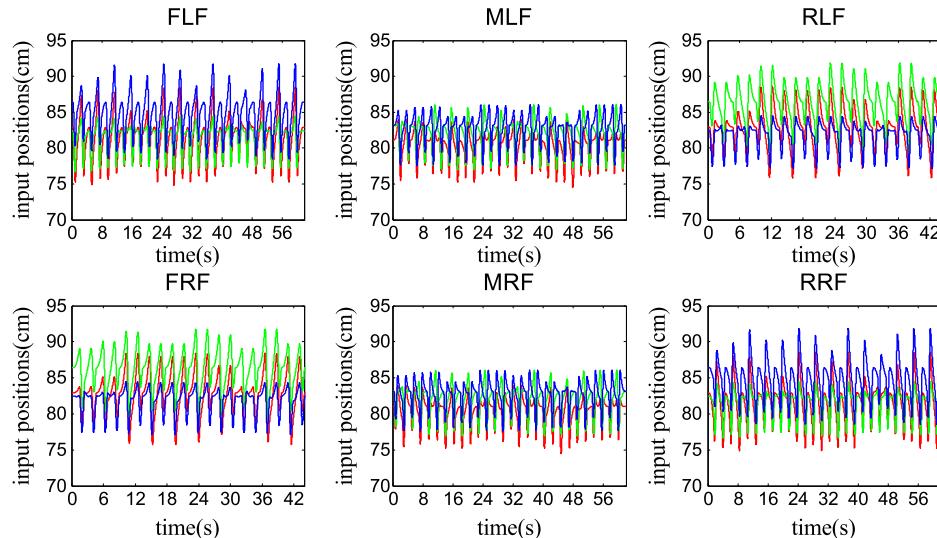
planned trajectory.  ${}^R_G T$  is the transformation matrix from the G-CS to the R-CS. It is obtained from the body motion computed in Section 4.3.  ${}^L_R T$  is the transformation matrix from the R-CS to the L-CS.  ${}^L_F T$  is the transformation matrix from the foot coordinate system(F-CS) to the L-CS. Formula (19) shows the detailed expression of  ${}^L_F T$ :

$${}^L_F T = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & \sin \alpha & p_i \cos \alpha \cos \beta \\ \sin \beta & \cos \beta & 0 & p_i \sin \beta \\ -\sin \alpha \cos \beta & \sin \alpha \sin \beta & \cos \alpha & -p_i \sin \alpha \cos \beta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

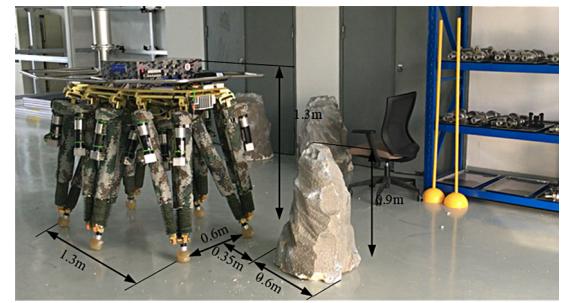
where  $\alpha$  and  $\beta$  are rotation angles of  $U_1$  along two vertical directions.  $\alpha$ ,  $\beta$  and  $p_i$  can be calculated from the following equations:

$$p_i = \sqrt{{}^L x^2 + {}^L y^2 + {}^L z^2} \quad (20)$$

$$\alpha = \arctan\left(\frac{{}^L z}{{}^L x}\right) \quad (21)$$



**Fig. 19.** Control inputs of the first experiment.



**Fig. 20.** The second experimental site (one obstacle closely in front of the robot).

$$\beta = \arcsin\left(-\frac{{}^L y}{p_i}\right) \quad (22)$$

${}^L_F T$  are computed by substituting  $\alpha$ ,  $\beta$  and  $p_i$  into formula (19). The other two inputs  $p_j$  and  $p_k$  are obtained from the following equations:

$$p_j = |{}^L \Gamma - {}^L \mathbf{U}_2| \quad (23)$$

$$p_k = |{}^L \Gamma - {}^L \mathbf{U}_3| \quad (24)$$

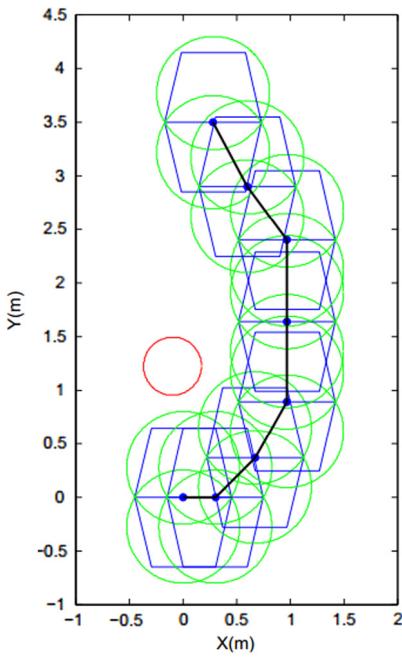
where  ${}^L \mathbf{U}_2$  and  ${}^L \mathbf{U}_3$  are positions of  $U_2$  and  $U_3$  in the L-CS.  $|\cdot|$  represents the length of the vector.  ${}^L \Gamma$  is calculated from the following equation:

$${}^L \Gamma = {}^L \mathbf{T}_G^R {}^R \mathbf{T}^G \Gamma. \quad (25)$$

Substituting foot trajectory  ${}^L \Gamma$  of each leg in formula (20), (23) and (24), all 18 input lengths of prismatic joints are computed.

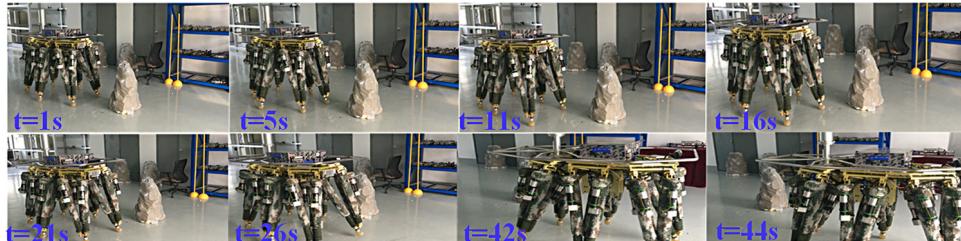
## 5. Experiment

The proposed obstacle avoidance and motion planning scheme is integrated on the Octopus-III robot. Several experiments are carried out and the results are presented in this section. Fig. 15 shows the control architecture consisting of obstacle detection and location, robot location, trajectory planning (Section 3) and gait generation (Section 4). To date, most robots use visual information from a stereo camera or distance information from a LIDAR (light detection and ranging) sensor to perceive the environment. For the

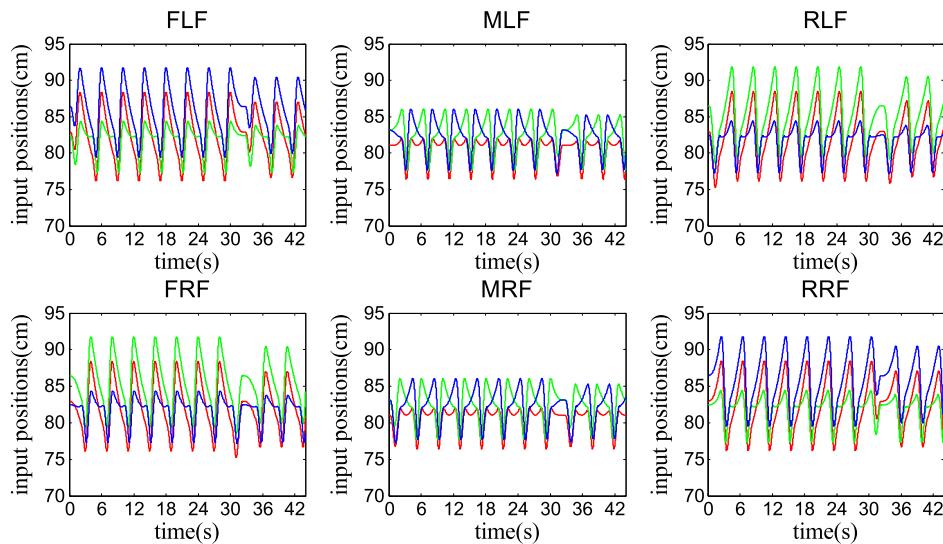


**Fig. 21.** The robot trajectory of avoiding one close obstacle.

indoor perception, a Microsoft Kinect V1 sensor is equipped. The FOV(field of view) of the Kinect V1 sensor is 57 degree horizontally and 43 degree vertically, and the detection range is 0.8 m to 4.0 m.



**Fig. 22.** Image flows of avoiding one close obstacle in front of the robot.



**Fig. 23.** Control inputs of the second experiment.

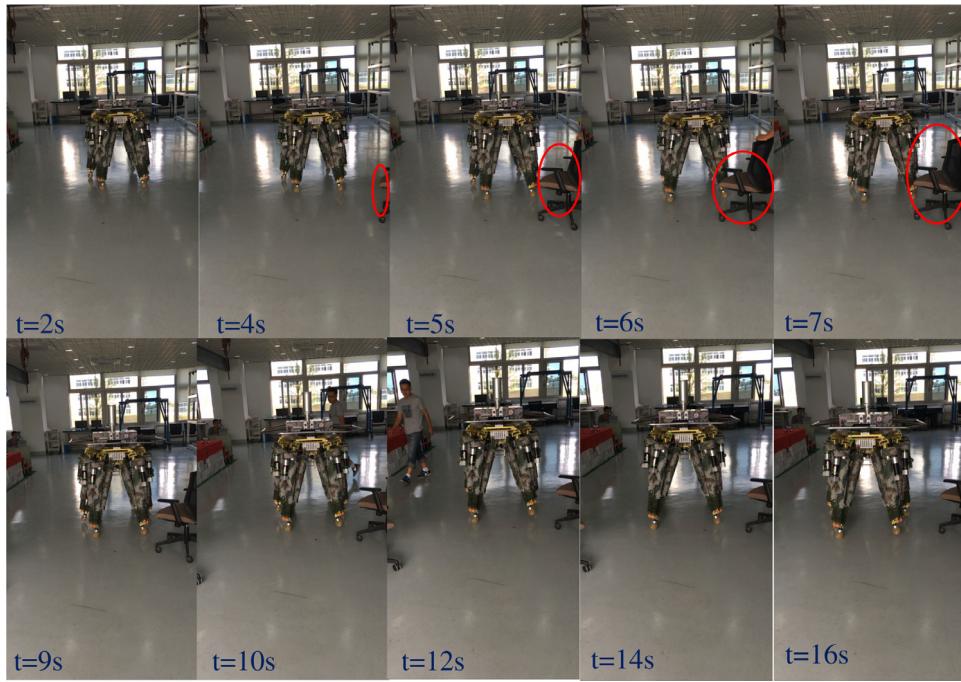
For the outdoor perception, a Velodyne PUCK LiDAR is equipped. The FOV of the Velodyne PUCK LiDAR is 360 degree horizontally and 30 degree vertically, and the detection range is up to 100.0 m.

A typical 2.5D grid map where each grid holds the object's height is built to detect the obstacle. Concretely, firstly the point cloud data  $\mathbf{P}^c(x^c, y^c, z^c)$ , which can describe the 3D spatial information of the real world, is obtained from the depth image captured by the Kinect or the distance information detected by the Velodyne LiDAR. And  $\mathbf{P}^R(x^R, y^R, z^R)$  in robot coordinates is transformed from  $\mathbf{P}^c(x^c, y^c, z^c)$  using the sensor's pose information relative to the robot. Then  $\mathbf{P}^R(x^R, y^R, z^R)$  is mapped into the corresponding grid based on the coordinates  $x^R$  and  $y^R$ . The height of the grid  $h$  is calculated as the average value of the coordinate  $z^R$ . The RANSAC algorithm is used to segment the ground, and the ground height is calculated at the same time. The grid is labeled as an obstacle by comparing its height  $h$  and the ground height.

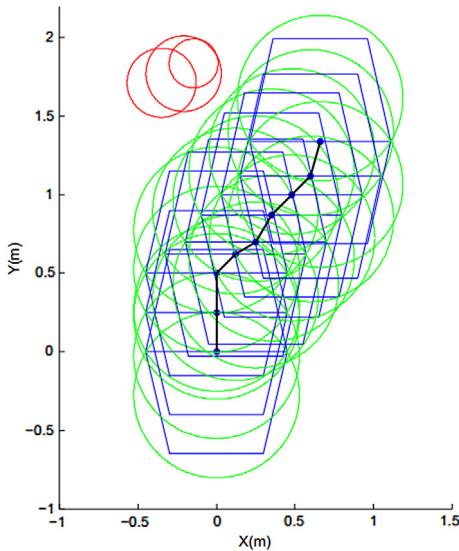
The site environment of the first experiment is illustrated in Fig. 16. Considering the site factors, the target position is specified as [0, 8], and the start position of the robot is [0, 0]. Considering the size of the robot Octopus-III, four obstacles are placed randomly. The minimum acceptable distance to the obstacle  $d_{\min}$  is 0.05 m, and the maximum limit distance  $d_{\max}$  is 0.2 m.

Fig. 17 shows the whole walking process of Octopus-III in the first experiment. The image flows clearly show that it reaches the target position successfully without colliding with any obstacles. The COG is located in the supporting triangle constructed by the feet all the time. The feet are regulated to move from the initial positions to the selected footholds.

Table 1 shows the detected obstacles' positions and Table 2 shows the recorded positions of the robot. Fig. 18 shows the robot trajectory in the first experiment. The detected obstacles are



**Fig. 24.** Image flows of avoiding the dynamic obstacle.



**Fig. 25.** The robot trajectory of avoiding the dynamic obstacle.

depicted by red circles, and the robot is represented by the blue hexagon. The black line represents the recorded robot trajectory. The robot passes through the obstacles and reaches the target position without collision. As mentioned above, the prismatic joints are actuated by motors to control the robot to walk along the planned trajectory. Fig. 19 shows all the 18 control inputs in the first experiment.

Fig. 20 shows the second experimental site. Only one obstacle is closely placed about 0.35m in front of the robot. The robot trajectory is shown in Fig. 21 with black lines. The red circle represents the detected obstacle, which is close to the robot. Fig. 22 shows the image flows. From Figs. 21 and 22, we can see that the robot first walks laterally, because the obstacle is too close for the robot to move forward directly. Then the robot selects the walking direction

**Table 1**  
The detected obstacles.

	x (m)	y (m)	Radius (m)
Obstacle 1	0.463	1.338	0.268
Obstacle 2	-1.027	3.236	0.198
Obstacle 3	0.538	4.769	0.224
Obstacle 4	-1.012	6.366	0.202

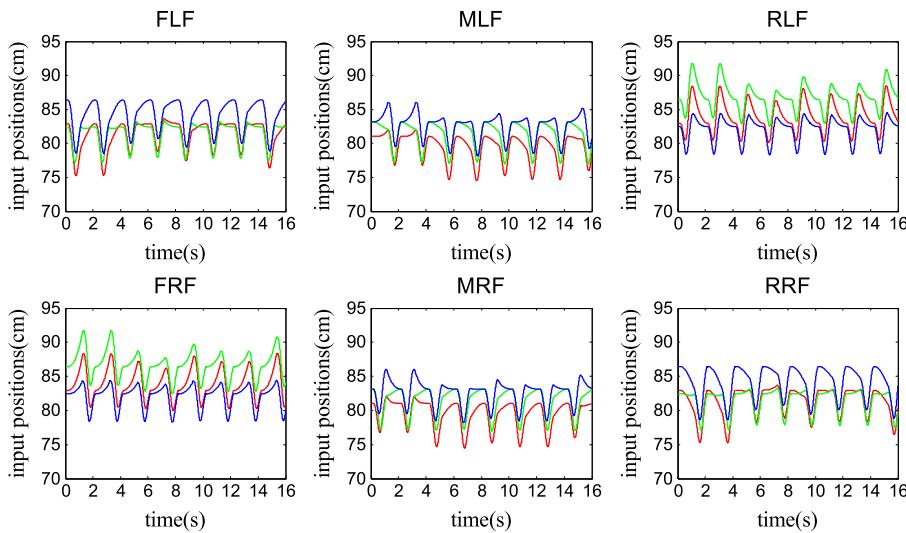
**Table 2**  
The recorded robot positions.

	x (m)	y (m)
Plan cycle 1	-0.371	0.371
Plan cycle 2	-0.546	1.023
Plan cycle 3	-0.489	1.749
Plan cycle 4	-0.314	2.401
Plan cycle 5	-0.014	2.920
Plan cycle 6	-0.012	3.669
Plan cycle 7	-0.312	4.189
Plan cycle 8	-0.487	4.841
Plan cycle 9	-0.312	5.493
Plan cycle 10	-0.012	6.013
Plan cycle 11	-0.008	6.761
Plan cycle 12	-0.003	7.509
Plan cycle 13	0.002	8.057

which balances reaching the target and avoiding the obstacle using the algorithm proposed in Section 3. Fig. 23 shows the control inputs of the experiment.

Fig. 24 shows the dynamic obstacle avoidance test. In the previous few seconds, there are no obstacles and the robot walks towards the target normally. Then a dynamic obstacle (a chair), which is depicted by red ellipse in Fig. 24, moves close to the robot. Fig. 25 shows the robot trajectory of avoiding the dynamic obstacle. The red circles are the dynamic obstacle detected by the robot. According to Figs. 24 and 25, there is no collision and the robot avoids the dynamic obstacle successfully. Fig. 26 shows the control inputs of avoiding the dynamic obstacle.

After above indoor experiments, an outdoor experiment is also carried out to test the method's application in larger space. The outdoor experiment also aims to test the robot's speed and agility



**Fig. 26.** Control inputs of avoiding the dynamic obstacle.



**Fig. 27.** The outdoor experiment of shuttling through multiple rods.

of avoiding obstacles. As shown in Fig. 27, multiple rods (depicted by red ellipses) are placed in a row, the robot needs to shuttle through each rod one by one and reach the target without colliding with any rod.

Fig. 28 shows image flows of shuttling through three rods, the detailed walking trajectory is depicted in Fig. 29. In this experiment, the LiDAR sensor is used for outdoor requirement. The robot detects the rods, plans the trajectory to avoid the rods autonomously and reaches the target position without collision successfully. The average consuming time for obstacle detection, trajectory planning and motion generation is less than 0.1 s. The robot walks about 5.97 m in 30 s for shuttling through three rods.

The average shuttling speed is about 0.2 m/s which is close to the robot's designed normal walking speed (0.3 m/s).

## 6. Conclusion

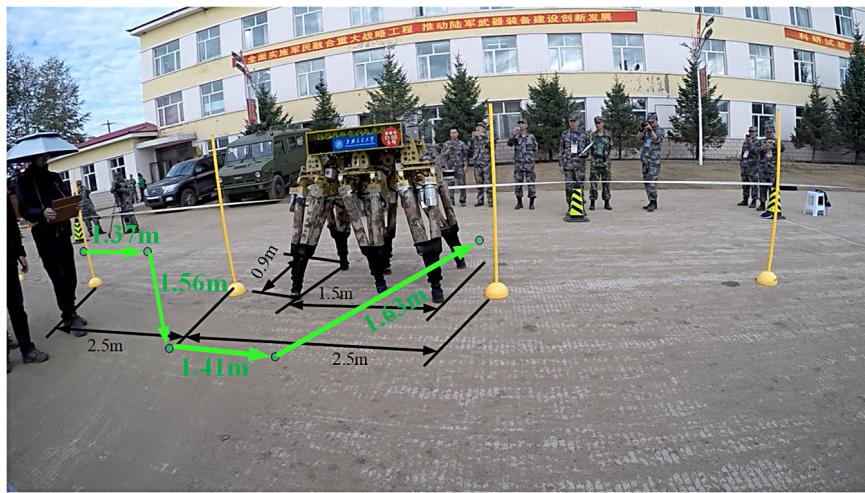
In this paper, we have proposed a novel obstacle avoidance and motion planning scheme for a hexapod robot Octopus-III. To the best of the authors' knowledge, seldom studies have worked on the related issue.

Compared with some hexapod walking robots like COMET-IV [37], RHex [5] and LittleDog [38], the major advantages of the algorithm this paper proposed are the superior mobility of the hexapod robot Octopus-III are taken advantage of in the path planning, and the trajectory planning method computes the walking direction by formulating the obstacle avoidance issue as an optimization problem. The drawbacks are the obstacles are simplified as circles which lacks terrain classification to decide walking gait. In the future work, more dynamical specialties will be taken account of and a 3-D map with details will be used to classify terrain and achieve smart navigation.

The theoretical contributions of this paper can be summarized as follows. The trajectory planning method computes the walking direction by formulating the obstacle avoidance issue as an optimization problem. The method takes advantage of the superior mobility of the legged robot and plans an optimum trajectory to the target position without colliding with any obstacles. What is more, the motion generation approach is developed to control the robot to walk along the planned trajectory. The approach coordinates



**Fig. 28.** Image flows of shuttling through rods.



**Fig. 29.** The detailed trajectory of shuttling through rods.

the body motion and the feet motions taking into accounts the constraints of the walking stability and the kinematic feasibility simultaneously. Finally, the proposed scheme is applied to the hexapod robot Octopus-III, and several experiments are carried out. The robot successfully avoids obstacles in different experiments. The experimental results show that the scheme works well in reality.

## Acknowledgments

This work was supported by the National Nature Science Foundation of China (Grant No. U1613208), and the Shanghai Science and Technology Committee (Grant No. 16DZ1201001).

## References

- [1] M.H. Raibert, Legged Robots that Balance, MIT press, 1986.
- [2] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, K. Fujimura, The intelligent ASIMO: System overview and integration, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, Vol. 3, IEEE, 2002, pp. 2478–2483.
- [3] M. Raibert, K. Blankespoor, G. Nelson, R. Playter, T. Team, Bigdog, the rough-terrain quadruped robot, in: Proceedings of the 17th World Congress, Vol. 17, no. 1, 2008, pp. 10822–10825.
- [4] C. Semini, N.G. Tsagarakis, B. Vanderborght, Y. Yang, D.G. Caldwell, HyQ-Hydraulically actuated quadruped robot: Hopping leg prototype, in: 2nd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, 2008, BioRob 2008, IEEE, 2008, pp. 593–599.
- [5] U. Saranli, M. Buehler, D.E. Koditschek, RHex: A simple and highly mobile hexapod robot, Int. J. Robot. Res. 20 (7) (2001) 616–631.
- [6] O. Stasse, B. Verrelst, B. Vanderborght, K. Yokoi, Strategies for humanoid robots to dynamically walk over large obstacles, IEEE Trans. Robot. 25 (4) (2009) 960–967.
- [7] P. Michel, J. Chestnutt, J. Kuffner, T. Kanade, Vision-guided humanoid footstep planning for dynamic environments, in: 2005 5th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2005, pp. 13–18.
- [8] L. Baudouin, N. Perrin, T. Moulard, F. Lamirault, O. Stasse, E. Yoshida, Real-time replanning using 3d environment for humanoid robot, in: 2011 11th IEEE-RAS International Conference on Humanoid Robots, (Humanoids), IEEE, 2011, pp. 584–589.
- [9] I. Havoutis, J. Ortiz, S. Bazeille, V. Barasuol, C. Semini, D.G. Caldwell, Onboard perception-based trotting and crawling with the hydraulic quadruped robot (HyQ), in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), IEEE, 2013, pp. 6052–6057.
- [10] J.R. Rebula, P.D. Neuhaus, B.V. Bonnlander, M.J. Johnson, J.E. Pratt, A controller for the littledog quadruped walking on rough terrain, in: 2007 IEEE International Conference on Robotics and Automation, IEEE, 2007, pp. 1467–1473.
- [11] A. Stelzer, H. Hirschmuller, M. Gorner, Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain, Int. J. Robot. Res. 31 (4) (2012) 381–402.
- [12] D. Belter, P. Skrzypczyński, Rough terrain mapping and classification for foothold selection in a walking robot, J. Field Robot. 28 (4) (2011) 497–528.
- [13] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, Int. J. Robot. Res. 5 (1) (1986) 90–98.
- [14] S.M. LaValle, Rapidly-Exploring Random Trees: A New Tool for Path Planning, 1998.
- [15] J. Borenstein, Y. Koren, Real-time obstacle avoidance for fast mobile robots, IEEE Trans. Syst. Man Cybern. 19 (5) (1989) 1179–1187.
- [16] J. Borenstein, Y. Koren, The vector field histogram-fast obstacle avoidance for mobile robots, IEEE Trans. Robot. Autom. 7 (3) (1991) 278–288.
- [17] I. Ulrich, J. Borenstein, VFH+: Reliable obstacle avoidance for fast mobile robots, in: 1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings, Vol. 2, IEEE, 1998, pp. 1572–1577.
- [18] V. Sezer, M. Gokasan, A novel obstacle avoidance algorithm: “Follow the gap method”, Robot. Auton. Syst. 60 (9) (2012) 1123–1134.
- [19] F. Hoeller, D. Schulz, M. Moors, F.E. Schneider, Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, IROS 2007, IEEE, 2007, pp. 1260–1265.
- [20] A.T. Rashid, A.A. Ali, M. Frasca, L. Fortuna, Multi-robot collision-free navigation based on reciprocal orientation, Robot. Auton. Syst. 60 (10) (2012) 1221–1230.
- [21] A. Cherubini, F. Chaumette, Visual navigation of a mobile robot with laser-based collision avoidance, Int. J. Robot. Res. 32 (2) (2013) 189–205.
- [22] Y. Yu Lwin, Y. Yamamoto, Obstacle-responsive navigation scheme of a wheeled mobile robot based on look-ahead control, Ind. Robot: Int. J. 39 (3) (2012) 282–293.
- [23] C.-H. Ko, K.-Y. Young, Y.-H. Hsieh, Optimized trajectory planning for mobile robot in the presence of moving obstacles, in: 2015 IEEE International Conference on Mechatronics, (ICM), IEEE, 2015, pp. 70–75.
- [24] Y. Zhu, T. Zhang, J. Song, X. Li, A new bug-type navigation algorithm for mobile robots in unknown environments containing moving obstacles, Ind. Robot: Int. J. 39 (1) (2012) 27–39.
- [25] H. Yang, X. Fan, P. Shi, C. Hua, Nonlinear control for tracking and obstacle avoidance of a wheeled mobile robot with nonholonomic constraint, IEEE Trans. Control Syst. Technol. 24 (2) (2016) 741–746.
- [26] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, M. Hutter, Navigation planning for legged robots in challenging terrain, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), IEEE, 2016, pp. 1184–1189.
- [27] N. Kumar, Z. Vámossy, Z.M. Szabó-Resch, Robot obstacle avoidance using bumper event, in: 2016 IEEE 11th International Symposium on Applied Computational Intelligence and Informatics, (SACI), IEEE, 2016, pp. 485–490.
- [28] K.-T. Song, S.-Y. Jiang, S.-Y. Wu, Safe guidance for a walking-assistant robot using gait estimation and obstacle avoidance, IEEE/ASME Trans. Mechatronics 22 (5) (2017) 2070–2078.
- [29] L. Tai, S. Li, M. Liu, A deep-network solution towards model-less obstacle avoidance, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS), IEEE, 2016, pp. 2759–2764.
- [30] L. Tai, M. Liu, Mobile robots exploration through cnn-based reinforcement learning, Robot. Biomimetics 3 (1) (2016) 24.
- [31] F. Tedeschi, G. Carbone, Hexapod walking robot locomotion, in: Motion and Operation Planning of Robotic Systems, Springer, 2015, pp. 439–468.

- [32] M. Petternella, S. Salinari, Simulation by digital computer of walking machine control system, in: IFAC, IIC, and ANIPLA, Symposium on Automatic Control in Space, 5th, Genoa, Italy, 1973.
- [33] R.B. McGhee, G.I. Iswandhi, Adaptive locomotion of a multilegged robot over rough terrain, *IEEE Trans. Syst. Man Cybern.* 9 (4) (1979) 176–182.
- [34] K. Nonami, Q.-J. Huang, Humanitarian mine detection six-legged walking robot COMET-II with two manipulators, in: Proceedings of the 4th International Conference on Climbing and Walking Robots, CLAWAR'2001, Karlsruhe, Germany, 2001, pp. 24–26.
- [35] A. Roennau, G. Heppner, L. Pfotzer, R. Dillmann, Lauron V: Optimized leg configuration for the design of a bio-inspired walking robot, in: Proc. Conference on Climbing and Walking Robots, CLAWAR, 2013.
- [36] G. Carbone, A. Di Nuovo, A hybrid multi-objective evolutionary approach for optimal path planning of a hexapod robot, in: International Workshop on Hybrid Metaheuristics, Springer, 2016, pp. 131–144.
- [37] K. Nonami, R.K. Barai, A. Irawan, M.R. Daud, Design and optimization of hydraulically actuated hexapod robot COMET-IV, in: Hydraulically Actuated Hexapod Robots, Springer, 2014, pp. 41–84.
- [38] P.D. Neuhaus, J.E. Pratt, M.J. Johnson, Comprehensive summary of the institute for human and machine cognition's experience with littledog, *Int. J. Robot. Res.* 30 (2) (2011) 216–235.



**Yue Zhao**, born in 1992, is currently a Ph.D. candidate at State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, China. His research interest is visual control of legged robots.



**Xun Chai**, born in 1990, is currently a Ph.D. candidate at State Key Laboratory of Mechanical System and Vibration, Shanghai Jiao Tong University, China. His research interest is visual control of legged robots.



**Feng Gao**, born in 1956, is currently a professor at Shanghai Jiao Tong University, China. His main research interests include parallel robots, design theory and its applications, large scale and heavy payload manipulator design, large scale press machine design and optimization, design and manufacturing of nuclear power equipment, legged robots design and control.



**Chenkun Qi**, born in 1978, is currently an assistant researcher at Shanghai Jiao Tong University, China. His main research interests include industrial process modeling and control, system identification modeling and control of distributed parameter systems modeling, control of robotics and intelligent learning.