Robotics and Autonomous Systems 59 (2011) 1036–1048

Contents lists available at SciVerse ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Improving traversability of quadruped walking robots using body movement in 3D rough terrains

Vo-Gia Loc [a], Ig Mo Koo [a], Duc Trong Tran [a], Sangdoek Park [b], Hyungpil Moon [a], Hyouk Ryeol Choi [a,*]

[a] School of Mechanical Engineering, Sungkyunkwan University, Chonch'on-dong, Jangan-gu, Suwon, Kyonggi-do, Republic of Korea

[b] Division of Applied Robot Technology, Korea Institute of Industrial Technology, Ansan, Republic of Korea

ARTICLE INFO

Article history:
Received 12 June 2011
Accepted 7 August 2011
Available online 25 August 2011

Keywords:
Legged robot
Quadruped robot
Body movement
Rough terrain

ABSTRACT

This paper presents a study on improving the traversability of a quadruped walking robot in 3D rough terrains. The key idea is to exploit body movement of the robot. The position and orientation of the robot are systematically adjusted and the possibility of finding a valid foothold for the next swing is maximized, which makes the robot have more chances to overcome the rough terrains. In addition, a foothold search algorithm that provides the valid foothold while maintaining a high traversability of the robot, is investigated and a gait selection algorithm is developed to help the robot avoid deadlock situations. To explain the algorithms, new concepts such as reachable area, stable area, potential search direction, and complementary kinematic margin are introduced, and the effectiveness of the algorithms is validated via simulations and experiments.

1. Introduction

Locomotion on the natural rough terrains where a biped or a wheeled vehicle cannot cope with is one of the most attractive issues for quadruped walking robots. A quadruped robot may use static or dynamic walking gaits to walk over rough terrains. With static walking, the vertical projection of the robot's center of gravity (written by CoG in this paper) needs to be always maintained inside the polygon formed by its supporting legs (termed as "support polygon") [1,2]. On the contrary, the CoG is allowed to be moved outside the support polygon temporarily in dynamic walking, and the positions of the foot placement heavily depend on the dynamics of the robot's body. However, the current control techniques on dynamic walking do not allow the robot to negotiate severe changes of terrain conditions [3,4].

The ability to overcome rough terrains, called *traversability* in this paper, has been the focus of much research on legged robots and how to improve traversability has been the issue widely investigated. McGhee et al. proposed basic parameters for the locomotion of legged robots such as forbidden cell, reachable area, and kinematic margin [1]. Hirose et al. developed a terrain-adaptive gait for the quadruped robot [5]. In their work, a crab angle allowing the robot to move with omni-direction and the concept of foothold search area were discussed. Then, fundamental problems of the quadruped walking robot in uneven terrains were explored [6–9]. Also, a pioneering work on using graph-based search was presented by Pal and Jayarajan [10]. In their work, some suitable states of the robot were considered among a limited number of robot's states. The graph-based approach was studied in more details by applying a hierarchical planning method or by a multi-step motion planning technique [11–13]. Meanwhile, Santos et al. applied a quadruped walking robot named SILO4, on a rough terrain composed of ditches and slopes [14]. In addition, some research reported how to improve the adaptability of hexapod robots in uneven terrains by using global path planners [15, 16]. Recently, the research on the locomotion of quadruped walking robots in the rough terrains are prevailing with the L2 (Learning Locomotion) project sponsored by DARPA (The Defense Advanced Research Projects Agency). The researchers involved in this project have explored comprehensive problems of quadruped walking robots in severely rough terrains. Essential problems on free gait were carefully discussed and advanced techniques were developed [17–23]. The achievements of the aforementioned works are highly appreciated, but still several problems need to be further investigated. The following are two major problems of previous studies.

The first problem is that body movement was not exploited to improve the traversability of the robot in rough terrains. Statically walking robots navigate rough terrains by sequentially moving body and swinging one leg at a time. However, previous researchers have solely investigated on the foothold search algorithm while almost ignoring the role of body movement in increasing the robot's traversability. In their works, body movement

* Corresponding author.
E-mail addresses: vogialoc@me.skku.ac.kr (V.-G. Loc), kooigmo@me.skku.ac.kr (I.M. Koo), jamestran@me.skku.ac.kr (D.T. Tran), sdpark88@gmail.com (S. Park), hyungpil@me.skku.ac.kr (H. Moon), hrchoi@me.skku.ac.kr (H.R. Choi).

0921-8890/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2011.08.007

has been used to minimize the traveled distance of the CoG [19], to obtain a better stability in rough terrain [21,23,24], to possess higher flexibility with a movable weighting device [25], to follow a planned trajectory [14], to have a continuous smooth motion [18,22] or to eliminate four-leg supporting phases [19,23]. In a highly rough terrain, the distance from the current foothold to the next desired foothold is normally large and it is not easy to place the next swing leg to the next desired foothold without drastically adjusting the robot's body posture. In such challenging terrains – that contain many large unsafe areas for placing the foot – the work that only concentrated on foothold search algorithm could find only some or even no terrain area to find the footholds. When the number of choices for evaluating decreases, the chance to find a suitable solution does, thus the robot's traversability is diminished. In such a case, it is necessary to find a way to increase the acceptable area for the robot to place the next swing leg at first, before trying to develop a foothold search algorithm. Fortunately, exploiting body movement of the robot to improve the traversability, is a solution for these rough terrains.

The second problem is that the previous foothold search algorithms mostly find a position to place the next foothold based on the quality of the geometric conditions around that position. Actually, the quality of the next foot position also closely relates to the positions of the other legs and the robot's posture. Selection of the next foothold solely based on geometric conditions may restrict the flexibility of the robot in upcoming walking cycles, thus reducing the robot's traversability. Some of recent reports have considered position of other feet, body postures in choosing the next foot placement [18,19,23]. However, the relation between the quality of the foothold and the elements was not clearly stated.

In contrast with the previous approaches, we sacrifice all such properties of the body [14,18,21–25] to concentrate on improving the traversability of the robot. The robot's body is systematically adjusted to maximize the reachability of the next swing leg. As a result, the traversability of the robot is improved. A foothold search algorithm that considers the current configuration of the robot in finding the next foot placements is also proposed. The algorithm is designed with simple geometric calculation to ensure the real-time requirement. Besides, a gait selection algorithm is proposed to help the robot deal with deadlock situations.

This paper is organized as follows. Section 2 briefly states the mentioned problems in a deeper level. In Section 3, notations and definitions to study locomotion of the robot in 3D rough terrains are proposed. The algorithm to find the orientation and position of the robot's body is presented in Section 4. The foothold search algorithm is presented in Section 5, and the gait selection is explained in Section 6. Then, analysis and results are shown in Section 7. Finally, conclusions are given in Section 8.

2. Problem statements

In this section, a model of a quadruped walking robot is introduced and two significant features affecting the traversability of the robot are addressed such as body movement and foothold selection. Discussions on using global and local planners in rough terrains are given as well.

2.1. A generic model of quadruped walking robot

A generic model of the quadruped walking robot is used in this work. It has four legs and each leg has three active degrees of freedom (DOFs) with three revolute joints, as illustrated in Fig. 1. The workspace L_i of the leg i is a part of a sphere whose center exists at the hip, and the radius of the sphere is the length of the leg when it is fully extended. In this work, the joint range of the first joint of the robot is assumed to be from 0° to 90°. Therefore, the workspace of a leg is a part of a quadrant of the sphere. Each workspace is assumed to have no overlapped region. The global coordinate frame is denoted as $O_G - xyz$ and the robot's body coordinate frame is represented as $O_R - x'y'z'$. The CoG of the robot is assumed to be located at the geometric center of the body and is represented by a vector $\mathbf{r}_G = [x_r\ y_r\ z_r]^T$ with respect to the global coordinate frame. As shown in Fig. 1, the foothold and the first joint (the center of the workspace) are represented by F_i and B_i ($i = 1, 2, 3, 4$), respectively.

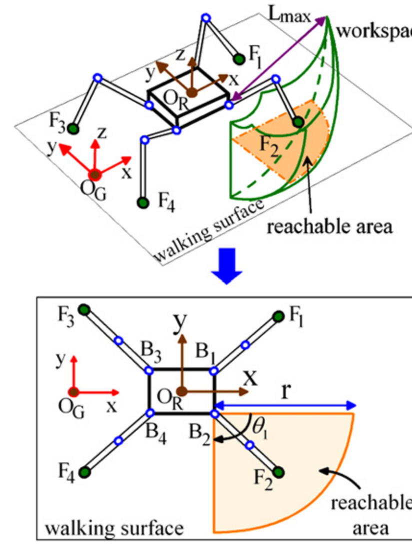


Fig. 1. Generic model of the quadruped walking robot and reachable area. L_{max} represents the length of the robot's leg at fully extended.

2.2. Idea on body movement and traversability

Definition 1. *Reachable area R_i of leg i is defined as the intersection of the leg's workspace with the terrain. It contains all terrain cells that the robot can reach with leg i while its body is fixed.*

$$R_i = L_i \cap T \tag{1}$$

where T represents the terrain model. T is divided into grid cells and represented by a height function such as

$$f : \mathbb{R}^2 \to \mathbb{R}, \quad f(x, y) = h \tag{2}$$

where h is the height of the terrain cell $T(x, y)$ and (x, y) indicates the location of the cell in the xy-plane. Consequently, the shape of the reachable area R_i is a quarter of a circle as shown in Fig. 1 and its size depends on the position and orientation of the robot's body. When the robot changes its body position and/or orientation, the location of the workspace and the reachable area also are changed. It leads to different reachability of the next swing leg with each different position and/or orientation of the robot's body. By exploiting this characteristic feature of the legged robot, the robot's traversability in the rough terrain can be improved very much. The idea is illustrated in Fig. 2 in which the next swing leg is leg 4. With the position of CoG as in Fig. 2(a), the reachable area of leg 4 is overlapped with the prohibited area. Here, the prohibited area corresponds to the areas that the robot cannot place its foot on such as obstacles, ditches, etc. Therefore, in such situation, the robot cannot swing the next swing leg over the prohibited area to move forward. However, with the same four foot positions, if the robot adjusts its body to the new posture as in Fig. 2(b), the next swing leg can reach to the safe area (any terrain cell that is not inside the prohibited area) behind the prohibited area. It shows the close relation between the traversability of the robot and body movement. The reachable area plays an important role in improving the traversability and thus, the issue of controlling the

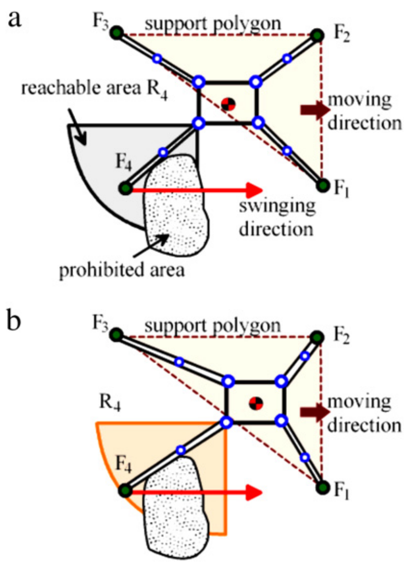


Fig. 2. Idea of exploiting the body posture of the robot to improve its traversability.

position and/or orientation of the robot's body is to have a suitable reachable area that has high reachability for the next swing leg.

2.3. Criteria in evaluating a foothold location

In most of previous work, the foot placement is determined by solely evaluating its geometric conditions. However, the current positions of the other legs should be considered in selecting the next foothold. It is because even a foothold with "good geometric conditions" may result in a "bad posture" of the robot that has low traversability due to the current positions of the other legs, and vice versa. The following example, illustrated in Fig. 3, explains that a low traversability posture can be immediately caused if only geometric property is cared in finding a foothold location. Note that this example is not a typical situation that always happens, but it is only for illustrating the case. In fact, the robot may meet a deadlock configuration in next several steps eventually because the traversability is reduced step-by-step. In this example, it is assumed that the next swing leg is leg 2, and the change of the reachable area R_i related to the changes of height and orientation is negligible. The cell F_2' is on a perfectly flat terrain and the cell F_2'' is on a non-flat terrain (for example, a slight slope or a bar). Both cells are far from the current foothold F_2 with the same desired stroke δ. Therefore, the geometric cost of the terrain cell F_2'' must be higher than that of the terrain cell F_2'. If the robot only cares about the geometric conditions in choosing the next foothold, it would swing a normal stroke δ from the current foothold F_2 to the cell F_2' as shown in Fig. 3(a) Unfortunately, the robot will meet a deadlock situation (a situation in which the robot cannot swing any leg further along the desired moving direction) due to very small movable area of the body in the next step if it places leg 2 to the cell F_2'.

The reason that swinging any leg is not possible is explained as follows. Let us call the distance from a foot i to the boundary of its reachable area R_i along the forward, backward and left, right sequently Δ_f^i, Δ_b^i, Δ_l^i, and Δ_r^i (see Fig. 3). To swing leg 3 forward, the robot must move its CoG inside *diagonal support quarter* (abbreviated as DSQ) F_1IF_2' or DSQ F_4IF_2' (when there is no swinging phase, the support polygon is divided into 4 DSQs by Diagonal Support Line 1 and Diagonal Support Line 2 as shown in Fig. 3). On the other hand, a quadruped robot moves its body (or CoG) a specific distance Δ along a given direction by moving all its legs with the same distance Δ along the inverse direction.

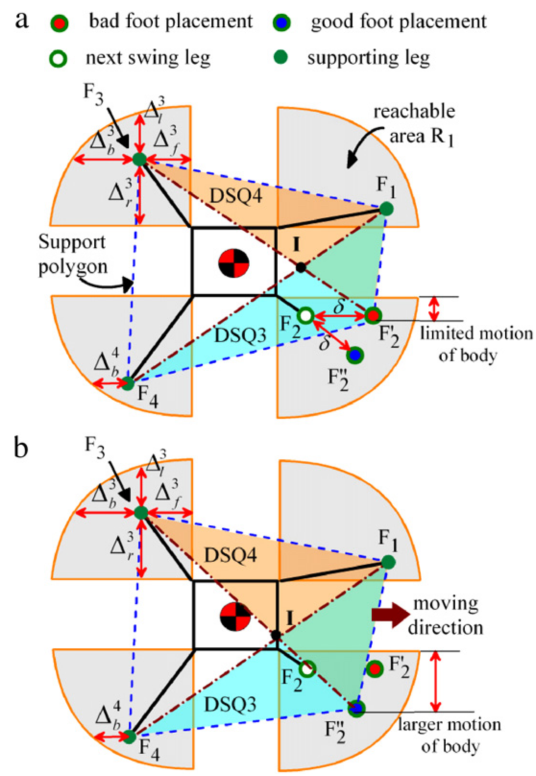


Fig. 3. A situation in which the robot may immediately meet a deadlock situation in the next step if the robot only cares about the geometric conditions in choosing the next swing legs.

Therefore, to move the CoG forward a distance Δ_f^B, to be inside DSQ F_1IF_2', the robot should move all its legs backward the same distance Δ_f^B. However, as shown in the figure, the maximum distance that leg 4 can move backward is only Δ_b^4, which is smaller than Δ_f^B. So, moving the CoG inside DSQ F_1IF_2' is not possible. Moving the CoG inside DSQ F_4IF_2' is also impossible due to restricted lateral motion of leg 2. Therefore, leg 3 cannot be lifted while keeping the equilibrium conditions. Similarly, the robot cannot move its CoG inside DSQ F_3IF_1 because of limited lateral motion of leg 1. Consequently, the CoG is locked inside DSQ F_3IF_4 and the robot cannot swing leg 4 without falling. This leaves only leg 1 as the next swing leg after leg 2 is swung. However, swinging leg 1 forward is also not possible because it is almost fully extended. Consequently, the robot is stuck in a deadlock situation. Fortunately, the robot can avoid this deadlock situation by placing leg 2 to F_2'' as shown in Fig. 3(b). In this posture, the robot's CoG is given a much larger movable area and the robot can swing leg 3 or leg 4 forward by adjusting the CoG inside the new support polygon $F_4F_1F_2''$ or $F_3F_1F_2''$. This example shows that evaluating a foothold location without considering the current configuration of the robot may deteriorate the traversability of the robot.

2.4. Global planner and local planner with legged locomotion over rough terrain

A global planner can avoid the local minima but it requires very good agreement between the model and the real robot during walking. Therefore, using a global planner to deal with a severely rough terrain is not practical because the difference between the real robot and the model is inevitable in such challenging terrains due to slippage and backlash. Meanwhile, a local planner cannot prevent the robot from local minima but it can fast react with the perturbation since it always re-plans the next step from the current position. Those problems can be dealt by using both global and local planning techniques. First, a sparse global planner is used to approximate an overall path from the initial position to the desired goal. Then, a local planner helps the robot walk along the planned path. Using both planners, the robot can avoid the local minimal and fast react with the perturbation, which are unavoidable in a rough terrain.

In this work, we only deal with a local planner that follows a pre-planned path generated by a global path planner and the terrain information is also assumed to be given. The local planner only uses the planned path as a reference path, it does not have to attempt to follow the path exactly. Therefore, searching footholds are not limited inside a small space around the planned path. The technique to find the global path and the way to get the map information are out of scope of this paper. The proposed local planner is composed of three algorithms which will be explained one by one in the following sections.

(1) *Body movement adjustment algorithm* (detailed in Section 4): The robot adjusts its body to a configuration that has a high possibility of finding a valid foothold for the next swing leg ("possibility" is represented by number of acceptable terrain cells inside a search region which will be mentioned later).
(2) *Foothold search algorithm* (detailed in Section 5): The robot finds a set of terrain cells that has high possibility of being the next foothold. Then, it evaluates geometric conditions of the cells inside the set of terrain cells in details to find the next foot placement.
(3) *Gait selection algorithm* (detailed in Section 6): If the robot cannot find a valid foothold for the next swing leg *i*, it must alternate the sequence of swinging the legs. The purpose of swinging the other legs and adjusting the body is to improve the reachability of leg *i*.

3. Exploring the interaction between a quadruped robot and a rough terrain

Before developing techniques to control the robot, it is necessary to have thoughtful understanding on how the robot interacts with a rough terrain. In this section, definitions and notations as new results from studying on the interaction is represented. All the concepts are applied for 3D rough terrains and some of them are represented in closed-form solutions.

3.1. Stable region and stable area of the robot's body

Definition 2. *Stable region* of the robot's body for a given body's orientation is a set of all body positions that satisfy the kinematic constraints and have a positive stable margin (denoted as SM in this paper and see [1] for its definition).

Definition 3. *Stable area* of the robot's body for a specific height with a fixed body's orientation is a set of all body positions that are inside the relevant stable region and have the same global *z*-coordinate.

The stable area of the body when leg *i* is swinging is denoted as S_i. Meanwhile, S is the stable area of the robot's body when all the legs are in the supporting phase.

A body position $\mathbf{O}_R = [x_r\ y_r\ z_r]^T$ is inside a stable area S_i with a given global z-coordinate z_C if it satisfies the following conditions

$$\mathrm{KM}_{\min}(x_r, y_r) \geq 0 \quad (3)$$

$$\mathrm{SM}(x_r, y_r) = 0 \quad (4)$$

$$z_r = z_C \quad (5)$$

where KM_i is the kinematic margin of the foot *i* (defined as [1]), and

$$\mathrm{KM}_{\min}(x_r, y_r) = \min(\mathrm{KM}_1, \mathrm{KM}_2, \mathrm{KM}_3, \mathrm{KM}_4) \quad (6)$$

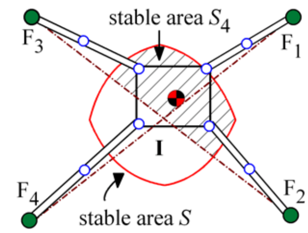


Fig. 4. Stable area S_i (cross region) is the intersection of the stable area S with the relevant support polygon.

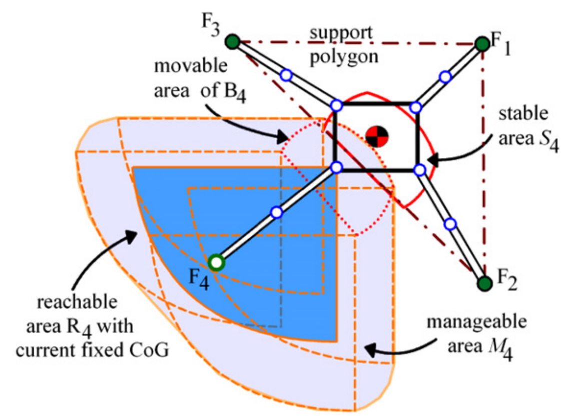


Fig. 5. Manageable area of a leg M_i is the Minkowski sum of the reachable area R_i with respect to the stable area S_i.

is the smallest kinematic margin, $\mathrm{SM}(x_r, y_r)$ is the stability margin of that body's position. The stable areas are determined depending on the kinematic limitations of the legs such as

$$\mathrm{KM}_{\min}(x_o, y_O) = 0, \qquad \mathrm{SM}(x_O, y_O) > 0, z_O, \quad z_O = z_C \quad (7)$$

and

$$\mathrm{KM}_{\min}(x_o, y_O) > 0, \qquad \mathrm{SM}(x_O, y_O) = 0, z_O, \quad z_O = z_C. \quad (8)$$

Illustration of the stable areas is shown in Fig. 4.

3.2. Manageable areas of legs and the robot

While the robot's body is moved inside the stable area S_i with a fixed orientation, the reachable area R_i is changed and covers an arbitrary area of the terrain. The coverage of the reachable area R_i is bounded inside an area, called *manageable area*. The manageable area for a given swing leg is defined as follows.

Definition 4. *Manageable area* M_i of a leg *i* for the specific height and orientation of the body means a set of all terrain cells that the leg can reach for all of the body's positions inside the relevant stable area S_i (see Fig. 5).

Therefore, M_i is the Minkowski sum of the reachable area R_i with the stable area S_i as follows [26].

$$M_i = R_i \oplus S_i. \quad (9)$$

Similarly, the manageable area of the robot is defined as follows.

Definition 5. *Manageable area of the robot*, M for the specific height and orientation of the body is the union of manageable areas of all legs such as

$$M = \cup M_i. \quad (10)$$

3.3. Concepts of desired set of foot positions

Upon a large manageable area M, the robot has a high flexibility of adjusting its body as well as many choices for selecting the next foothold. Obviously, the size of M relates to the height and

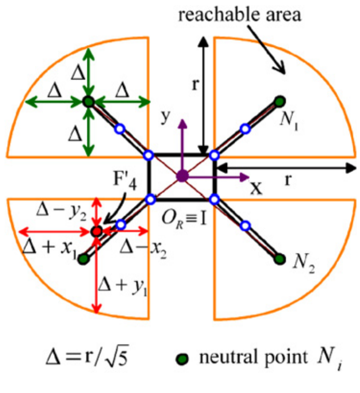


Fig. 6. Illustration of the desired set of foot positions.

orientation of the robot's body (the relation will be discussed later in Section 4). In addition, it also depends on the current relevant positions of the feet. It is necessary to find a set of relevant foot positions that is associated with the largest manageable area M with a given body's height and orientation. That set of foot positions (F_1, F_2, F_3, F_4) is called *desired set of foot positions*.

The desired set of foot positions is determined based on the following property. When the CoG is coincident with the intersection I of the two diagonal lines F_1F_4 and F_2F_3, the feet are at the neutral point N_i of their reachable area R_i (neutral point of a reachable area is the point that the distances from it to the boundary of the reachable area along x- and y- direction are the same. See Fig. 6).

Proposition 1. *Desired set of foot positions has the largest manageable area of the robot M for a given body's height with a fixed body's orientation.*

Proof. It is necessary to show that any set of foot positions which is not the desired set of foot positions has the smaller manageable area of the robot. Without loss of generality, let us assume that leg 4 is placed at N'_4 which is not a neutral point as shown in Fig. 6. Then, it needs to prove

$$M(N'_4) < M(N_4) \tag{11}$$

where $M(N'_4)$ is the manageable area of the robot when leg 4 is placed at N'_4, and $M(N_4)$ is the manageable area when leg 4 is placed at the neutral point N_4.

With the desired set of foot positions, the distances from any foot to its boundary of the reachable area along the x- and y-direction are the same: $(\Delta, \Delta, \Delta, \Delta)$. Therefore, the distances that the body can move from the intersection I forward, backward and left, right are also $(\Delta, \Delta, \Delta, \Delta)$ (if the body is moved a distance that is bigger than Δ, a foot will be out of its workspace as explained in Section 2.3). With a set that is not the desired set of foot positions, the movable distances of the body are smaller than that of the desired set. As shown in Fig. 6, the distances from N'_4 to the boundary of the reachable area R_4 along the x- and y-direction will be $(\Delta + x_1, \Delta - x_2, \Delta + y_1, \Delta - y_2)$. Consequently, the distances that the CoG can move from the intersection I backward is $(\Delta - x_2)$ and right is $(\Delta - y_2)$ (limited by mechanical limitation of leg 4), and from I forward and left is the same Δ (limited by mechanical limitation of the other leg). Therefore, the distances that the robot's CoG can move from I forward, backward and left, right are $(\Delta, \Delta - x_2, \Delta, \Delta - y_2)$. It leads to a smaller stable area (in comparison with the stable area of the desired set of foot positions) as shown in Fig. 7 since the stable area is determined by four arcs which are the kinematic limitation of each leg. As a result, we have

$$S(N'_4) < S(N_4) \tag{12}$$

where $S(N'_4)$ is the stable area of the robot when leg 4 is placed at N'_4 and $S(N_4)$ is the stable area when leg 4 is placed at the neutral point N_4 With the same height and orientation of the body, the size of the reachable area R_i is fixed. Therefore, from Eq. (9), when $S(N'_4) < S(N_4)$, it leads to a smaller size of $M(N'_4)$ compared to $M(N_4)$ as desired. □

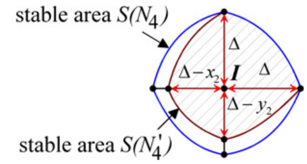


Fig. 7. Illustration of reduction of the stable area when a leg is not placed at the neutral point. Any set of foot position which is not the desired set of foot position will have smaller stable area than that of the desired set.

By maintaining the feet around the desired set of foot positions that is associated with the largest M, the robot always has a high traversability to deal with the reduced traversability that is mentioned in Section 2.3. Note that the conventional way of staggering the legs proposed by [2] shows higher traversability along the straight forward direction with a fixed order of swinging leg. However, the desired set of foot positions is preferred due to its high adaptability in omni-directional moving with any swinging order.

Also, *recovery foothold* and *recovery swing direction* are proposed to control the foot positions around the set of desired foot positions.

Definition 6. *Recovery foothold* is the foothold of the next swing leg that leads the legs' positions to the set of desired foot positions in the next walking cycle.

Definition 7. The swing direction of the next swing leg to place the foot to the recovery foothold is called the *recovery swing direction*.

During the travel, the robot always compares its posture in each walking cycle to maintain its footholds around the set of desired foot positions to ensure a high ability of avoiding deadlock situations.

3.4. Potential search direction

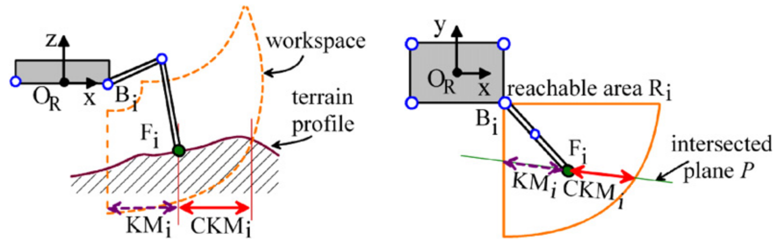
In a flat terrain, the robot just needs to swing its leg toward the moving direction (it is also the recovery swing direction in this case). However, in a rough terrain, it may not be possible to swing along the recovery swing direction due to obstacles because the robot must step its foot on a safe terrain cell outside the obstacles. Since a comprehensive evaluation of all local terrain cells takes an expensive calculation, it is faster to search for the next desired foothold along a direction that gives a high possibility of finding a valid foothold. Furthermore, the direction should not deviate too far from the recovery swing direction so that the robot's posture can be maintained around the set of desired foot positions.

Definition 8. *Potential search direction* is defined as the direction of searching the next foothold that is nearest to the recovery swing direction (the angle between the potential search direction and the recovery swing direction should be minimized) and gives sufficient possibility of finding a valid foothold. Sufficient depends on the configuration of the robot and the terrain type. It is represented by a number of "acceptable" terrain cells inside a specific search area that will be discussed later.

3.5. Complementary Kinematic Margin

To represent reachability of a leg, *Complementary Kinematic Margin* (CKM) which is opposed to *kinematic margin* (abbreviated as KM [1]) is defined as follows.

Definition 9. *Complementary Kinematic Margin* is defined as the projected distance on the xy-plane that the foothold of a leg can



(a) Projection on intersected plane P. (b) Top view.

Fig. 8. Illustration of Complementary Kinematic Margin and Kinematic Margin. (a) Shows the projection of the model on the plane P which passes through the given direction and is perpendicular to the xy-plane. (b) Shows the projection of the model on the xy-plane.

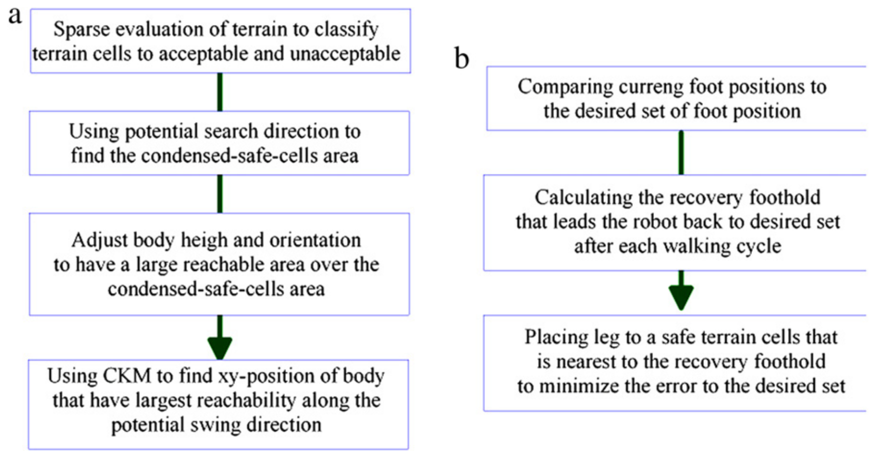


Fig. 9. (a) Shows steps to generate strategy 1 and (b) explains sequences of strategy 2.

travel in a given direction before reaching the boundary of the reachable area. As shown in Fig. 8 CKM exactly represents the reachability of a leg at the current robot's stance along a given direction, or how far the leg can swing along the direction.

The sequences of using the notations and definitions to exploit body movement to increase the reachability of the next swing leg while maintaining a posture that has a high traversability are briefed in Fig. 9.

4. Body movement adjustment algorithm

The principle of adjusting body is to have a large reachable area that contains many acceptable terrain cells inside for choosing the next foothold of the next swing leg. A minor change in the orientation and/or the position of the body leads to a change of size as well as shape of the reachable area. In this section, the adjustment of parameters of the robot's body in order to have a desired reachable area is presented.

4.1. Determination of desired body's orientation

– *Desired yaw angle of the body*: Adjusting the yaw angle of the robot's body may change the size and shape of the reachable area R_i of the next swing leg, and thus affects to the traversability. However, in a normal walking situation, using the yaw angle to track the pre-planned path has higher priority. Like the behavior of a living animal, the robot should point its head toward the current walking direction. Therefore, the desired yaw angle is calculated as the angle that leads the robot back to the pre-planned path in the next cycle. The equation to calculate the yaw angle that the robot should rotate in the next step is given by

$$\gamma_D = \gamma(t + \Delta_t) - \gamma(t) \tag{13}$$

where $\gamma(t + \Delta_t)$ is the angle between the local pre-planned path in the next step and the x-axis of the global frame. $\gamma(t)$ denotes the current yaw angle of the body with respect to the global frame and Δ_t is the time of a normal walking cycle.

The robot only uses the yaw angle to improve the traversability when it cannot find any solution to overcome the encountered terrain. Calculating the desired yaw angle in this case depends on the shape of the reachable area of the given robot. With the model used in this work, calculating the desired yaw angle is similar to that of the potential search direction because it brings more acceptable terrain cells inside the reachable area.

– *Desired roll and pitch angles of the body*: Rotation of the robot's body according to the roll and pitch angles leads to different reachable areas of the legs. In a normal walking, the size and the shape of all the reachable areas R_i should be the same to ensure similar behavior of the legs. Once again, the reachable area is the intersection of a part of a sphere and a terrain surface. Therefore, the similarity of the reachable areas R_i can be obtained when the robot's body is kept in parallel with the surface of the local terrain. In contrast, roll/pitch inclination between the robot's body and the surface of the local terrain leads to different size and shape of the reachable areas of the legs. Fig. 10 illustrates the difference of reachable areas of fore and hind legs when the pitch angle of the robot with the local terrain is α. The parameters in the figure are calculated as

$$l_1 = \sqrt{L_{\max}^2 - h_1^2 \cos^2(\alpha)} - h_1 \sin(\alpha) \tag{14}$$

$$w_1 = \sqrt{L_{\max}^2 - h_1^2} \tag{15}$$

$$l_2 = \sqrt{L_{\max}^2 - h_2^2 \cos^2(\alpha)} + h_2 \sin(\alpha) \tag{16}$$

$$w_2 = \sqrt{L_{\max}^2 - h_2^2}. \tag{17}$$

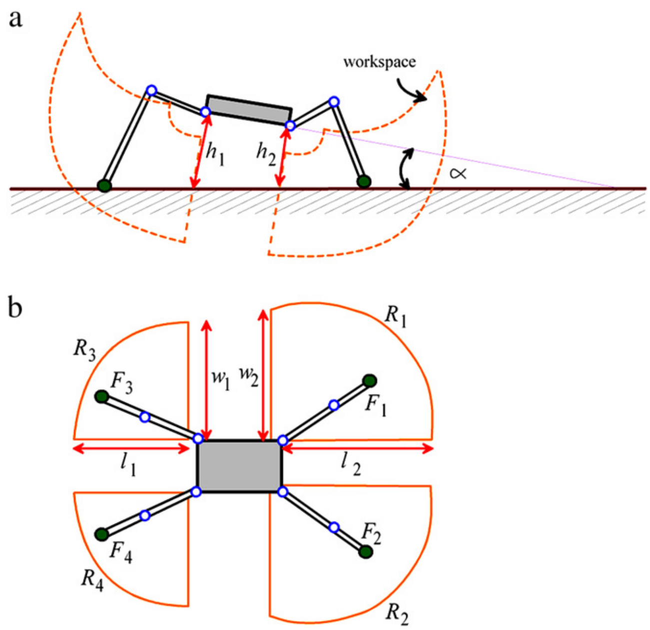


Fig. 10. Inclination of the robot's body with the surface of the local terrain leads to different reachable areas of fore and hind legs.

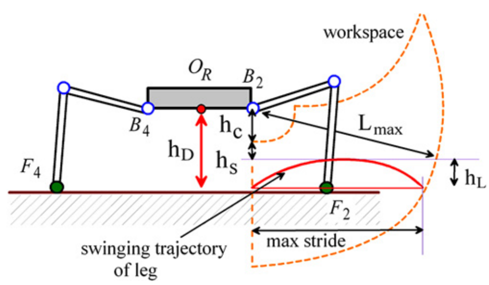


Fig. 11. Desired height of a quadruped robot to perform a high traversability. The robot tries to keep its body low to have largest reachable area while considering swinging trajectory as well as a safe distance from the boundary of the workspace.

Therefore, in normal walking, the robot's body is maintained parallel to the walking terrain. Consequently, the pitch angle and desired roll angle that the robot should rotate its body in the next step are calculated as follows

$$\alpha_D = \alpha(t + \Delta_t) - \alpha(t) \tag{18}$$

$$\beta_D = \beta(t + \Delta_t) - \beta(t) \tag{19}$$

where $\alpha(t + \Delta_t)(\beta(t + \Delta_t))$ is the approximated pitch (roll) angle of the surface of the local terrain in the next step with respect to the global frame. $\alpha(t)(\beta(t))$ is the current pitch (roll) angle of the body with respect to the global frame.

4.2. Determination of desired body's height

As mentioned, the reachable area is the intersection of a part of a sphere and the terrain. Its size is significantly changed depending on the body's height because the height is the distance from the center of the sphere to the intersection. Obviously, the intersection reaches its maximum size when the distance from the center of the sphere to the terrain, or the body's height, is smallest. As a result, the reachable area is maximized when the body's height is minimized. Moreover, the robot has its maximized stride length with the largest reachable area as shown in Fig. 11. Maximum reachable area and stride length assure a high traversability for the robot. Therefore, the desired height of the robot in the next step is calculated as

$$h_D = h_C + h_S + h_L \tag{20}$$

where h_C is the minimum reach of the leg (limited by joint constraints of the robot). h_S represents a safe distance to prevent from out of workspace and h_L is the lift up distance of the leg in the swinging phase as shown in Fig. 11.

4.3. Determination of desired body's xy-position

The desired *xy*-position of the body can be found by observing the change of the reachable area of the next swing leg. The principle is to move the body to a new position inside the stable area S_i where the reachable area R_i covers the terrain cells that have high possibility of being the next foothold.

To easily determine the recovery foothold in each step, the robot is allowed to start moving with the desired set of foot positions. Therefore, the desired recovery foothold of the next swing leg is simply determined by adding a normal stroke length to the current foothold along the moving direction. After one cycle of walking, the robot will be back to the desired set of foot positions. When the robot cannot place its next swing leg to the recovery foothold, the error between the real foot placement and the recovery foothold will be recorded and then compensated in calculating the recovery foothold of that leg in the next walking cycle.

After having the recovery foothold, the robot evaluates the geometric conditions around it more precisely. If the terrain conditions around the recovery foothold are good, the robot chooses the recovery foothold as the next foothold. The next *xy*-position of the CoG will be the position that has largest CKM along the recovery direction. Choosing the new position of the CoG like this ensures the reachability of the next swing leg to the desired foothold (also the recovery foothold in this case) is maximized.

If the recovery foothold is on a dangerous terrain area or out of reachability of the next swing leg, the robot must find another cell to place its next swing leg. Since it is not easy to evaluate all the local terrain cells to find the next desired foothold in real-time, the robot finds a direction that ensures a high possibility of finding a valid foothold—the potential search direction. Then, it maximizes the reachability of the next swing leg along that direction. Consequently, the possibility of finding a valid foothold is enhanced. When the next desired foot placement is not the recovery foothold, the sequence of finding the new *xy*-position of the CoG is explained as follows.

4.3.1. Sparse evaluation of the terrain

First, the quality of each cell of the given terrain T is pre-evaluated when the robot is set up. A terrain cell is classified into an acceptable cell or unacceptable cell based on geometric conditions of its own and its neighboring cells. An acceptable cell $T(x, y)$ must not near the boundary of obstacles, not on a deep hole or on a high obstacle, and not on a steep slope. These steps are performed only once when the robot is set up. Its calculation time does not interfere to the real-time performance of the robot.

4.3.2. Search region and possibility of finding a valid foothold

Up to this step, many cells inside the reachable area (for example, the cells are behind the current foothold) certainly have a low possibility of being chosen as the next foothold. To focus on the high potential cells, a search region is proposed instead of using the reachable area in finding the potential search direction. The search region is a trapezoidal region around the search direction as shown in Fig. 12. The trapezoidal shape ensures limitation of search direction and prevents from too small stride length swinging.

Besides, the possibility of finding a valid foothold along a search direction is represented by the *number of acceptable terrain cells*

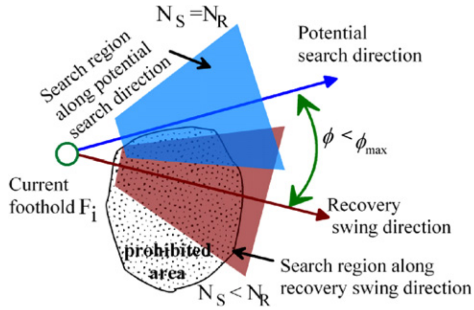


Fig. 12. Searching around the recovery swing direction to find the potential search direction. The search direction is varied around the recovery swing direction until there are enough numbers of acceptable terrain cells inside the trapezoidal search region.

inside the search region N_S along that search direction. With a higher number of acceptable terrain cells inside a search region, the possibility of finding a suitable cell as the next foothold is increased. The possibility is said "sufficient" if the number of acceptable terrain cells inside the search region N_S is bigger than a *required number of acceptable terrain cells* N_R (exact number depends on the roughness of the terrain and the robot's configuration).

4.3.3. Steps to find the potential search direction

– *Checking possibility of finding a valid foothold along the recovery swing direction*: If $N_S > N_R$ along the recovery swing direction, the possibility of finding a valid foothold along the recovery swing direction is large enough. In this case, the potential search direction becomes the same as the recovery swing direction. If the possibility of finding a valid foothold around the recovery swing direction is small ($N_S < N_R$), the robot has to find another search direction that will give sufficient possibility of finding a valid foothold.
– *Varying the recovery swing direction to find the potential search direction*: While $N_S < N_R$, the robot varies the search direction (and thus moves the search region) step by step around the recovery swing direction until $N_S \geq N_R$. To keep the potential search direction near the recovery direction, the search angle ϕ (the angle between the search direction and the recovery direction) is limited from $-\phi_{\max}$ to $\phi_{\max}$ where $\phi_{\max}$ is the maximum angle of search. If the robot cannot find any solution around the recovery swing direction (N_S is always smaller than N_R when $\phi \in [-\phi_{\max}, \phi_{\max}]$), there is a large obstacle in front of the robot. In that case, the robot just chooses the recovery swing direction as the potential search direction.

4.3.4. Using CKM to find the new body's xy-position

The robot scans all the positions inside the stable area S_i to find the next *xy*-position of the CoG. The position with the largest CKM_i of the next swing leg *i* along the potential search direction will be chosen because the search direction ensures a high possibility of finding a valid foothold. Moreover, the possibility is enhanced when the leg can reach its highest reachability (or to have the largest CKM) along the potential search direction. At times, many *xy*-positions of the COG inside S_i may have the same CKM_i, and the position with the shortest moving distance is the solution.

5. Foothold search algorithm

5.1. Potential foothold and set of candidate footholds

The set of candidate footholds in a step is still quite large. It is a problem with real-time control criteria when we want to consider many properties of a candidate such as height, slope, deviation, roughness, etc. So, we continue to remove the terrain cells that have low possibility of to be chosen as the next foothold. To do it, potential foothold and set of potential foothold are proposed.

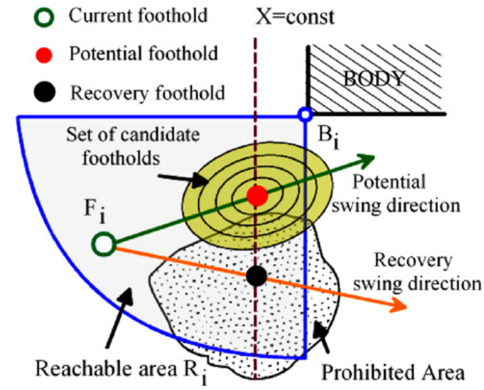


Fig. 13. Illustration of the potential foothold and the elliptical set of candidate footholds.

Definition 10. *Potential foothold* is the foothold that has the highest possibility to be chosen as the next leg placement.

Inspired from the idea that the potential foothold is the best foothold for the next leg placement and any deviation from this potential foothold should be minimized [17], a set of candidate footholds is proposed.

Definition 11. In this work, a *set of candidate footholds* is defined as a set of terrain cells which are inside an elliptic region whose center is the potential foothold.

The shape of the set is an ellipse instead of a circle because a deviation along the swing direction does not affect to the main locomotion direction as oppose to a deviation perpendicular to the swing direction does. For simplicity, the aspect ratio of the height and width of the ellipse is determined to be equal to 1:2. The concept of the set of foothold candidates is depicted in Fig. 13.

5.2. Steps to find the next foothold

5.2.1. Determination of potential foothold

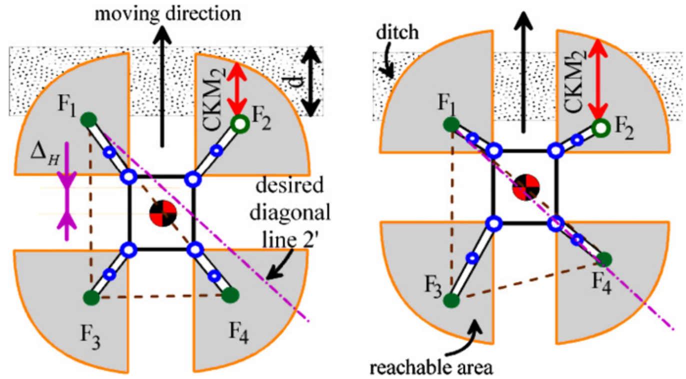
The error between the desired set of foot positions and the current footholds occurs mainly along the moving direction because the robot swings its leg along the moving direction to move forward. Therefore, the potential foothold is determined as the projection of the recovery foothold to the potential swing line along the direction that is perpendicular to the moving direction as shown in Fig. 13. The proposed method ensures the smallest error between the current foot positions and the desired set of foot positions along the moving direction.

5.2.2. Finding the next foothold inside the set of foothold candidates

The next foot placement is found by a spiral search from the potential foothold inside the elliptical set of candidate footholds. The cost function contains only one parameter, that is the distance from the evaluating cell to the potential foothold. The safe cell which is found first is chosen as the next foothold. A cell is safe to place its leg if it is reachable, collision-free swinging, and has a small enough deviation, and is not on a steep slope.

6. Gait selection

In this section, a new way of choosing the next swing leg is proposed. The principle is to increase reachability of the leg that needs to swing a large stride. Explanation of the algorithm is illustrated in Fig. 14, in which the body and the leg are just limited to move along the *x*-direction for a clearer image.



(a) $\text{CKM}_2 < d$. (b) $\text{CKM}'_2 > d$.

Fig. 14. Illustration of alternating swinging order to increase reachability of a desired leg.

6.1. Standard swinging sequence and previous approach

McGhee has pointed out in his work that among several sequences of switching leg, the sequence 4–2–3–1 is the most stable gait for continuous crawling gait [2]. Therefore, the sequence 4–2–3–1 is also called standard sequence and is used as the basic lifting sequence for most of the previous works as well as this work. However, in many cases, the robot will not be able to find a safe and reachable terrain cell to place its next swing leg that obeys a fixed sequence of swinging leg. A solution to escape from this deadlock situation is to choose another leg to swing in advance. In most of previous researches, the leg with smallest kinematic margin (KM) is chosen as the next swing leg [8,9,14]. It may somewhat prevent the robot from a deadlock situation. However, it does not directly relate to reachability of legs of the traversability of the robot.

6.2. Insufficient CKM—the reason breaking the standard sequence

The main reason that the robot cannot maintain the standard sequence is insufficient of reachability of a desired swing leg. The problem is illustrated in Fig. 14 where the robot is trying to overcome a ditch that its length is d. In this example, without lost of generality, the next swing leg is assumed to be leg 2. With the robot's configuration as shown in Fig. 14(a), the reachability of leg 2 is small, $\text{CKM}_2 < d$, the robot cannot swing leg 2 over the ditch with the current positions of the other feet. In this case, the robot must swing other legs in advance to find a way to avoid the situation. In other words, the reason breaking the standard sequence is insufficient of the desired swing leg. Moreover, choosing and swinging other legs must be considered how to give enough reachability for the leg that breaks the standard sequence, or the leg that need to have large reachability.

6.3. Accumulating CKM for the leg that needs to have large CKM

As in Fig. 14(a), CKM of leg 2 is CKM_2 is smaller than the ditch's length d. To be able to swing leg 2 over the ditch, CKM of leg 2 must be bigger than d or CKM of leg 2 must be at least increased a quantity $\Delta_H > (d - \text{CKM}_2)$. On the other hand, the way to increase CKM of leg 2 is to move the reachable area R_2 forward. It can be done by moving the robot's body forward and the distance that the body should be moved forward to have CKM of leg 2 larger than d is calculated as

$$\Delta_H > (d - \text{CKM}_2) \tag{21}$$

because the distance the reachable area is moved forward is equal with the increase of CKM of legs. To do it, the gait selection algorithm alternates the leg lifting sequence and swings the hind legs at first. As shown in the figure, the new foothold of leg 4 must be behind the desired *diagonal line* 2′ to allow the body move forward as far as the distance Δ_H without falling. On the other hand, the distance Δ_H that the robot's body moves forward is equal to the increase of the CKM of the fore legs. Therefore, CKM'_2 is

$$\text{CKM}'_2 = \text{CKM}_2 + \Delta_H. \tag{22}$$

Obviously, $\text{CKM}'_2 > d$ or the robot can swing leg 2 over the ditch as shown in Fig. 14(b). Therefore, by choosing leg 4 as the next swing leg in advance, the robot can increase the reachability of the fore legs, or improve the robot's traversability. The similar phenomenon happens with hind legs and the same principle can be applied.

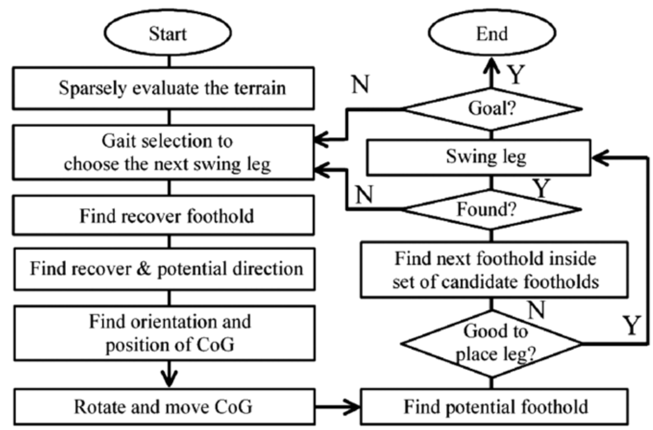


Fig. 15. Overall flowchart of the proposed free gait.

Fig. 15 shows the overall flowchart of the algorithm. When alternating the leg lifting sequence also cannot find the solution for the deadlock situation, the robot will try with the other body's orientation (this step is not shown in Fig. 15 for a clearer picture). If there is still no solution, the local environment is considered as impassable, the global path planner needs to provide a new route.

7. Results

In this section, the proposed algorithms were implemented to a sprawl posture quadruped robot, named MRWALLSPECT IV (abbreviated as MR4 and is upgraded from MRWALLSPECT III robot [27]) as shown in Fig. 16. The robot has three-DOF active joints for each leg. The length of its leg is 550 mm with 226 mm of thigh and 226 mm of tibia. The length of its body is 226 mm.

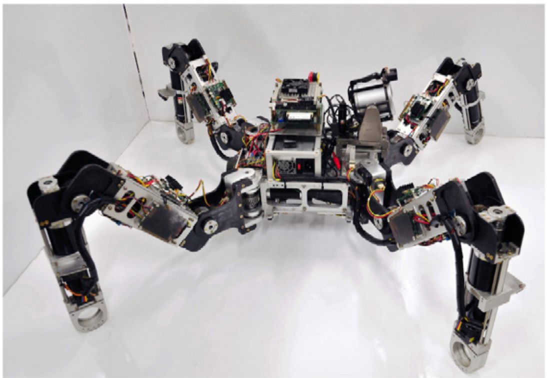


Fig. 16. MRWALLSPECT IV robot.

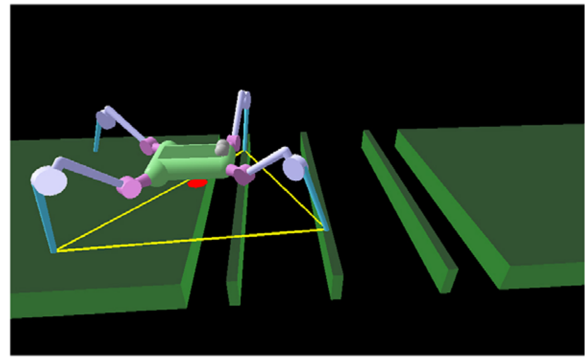


Fig. 18. The environment used in the comparison.

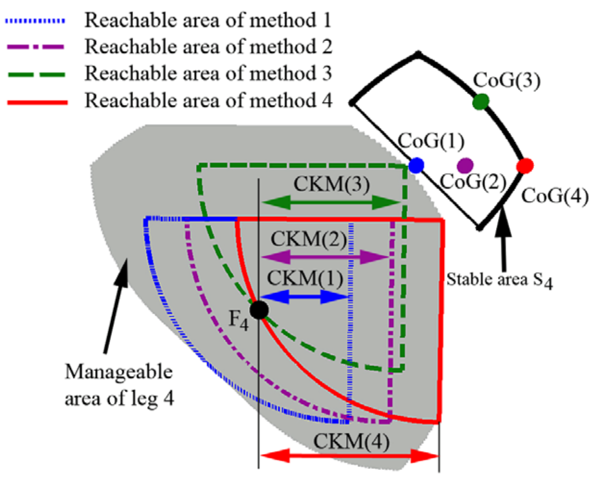


Fig. 17. Stable area of body and manageable area of the next swing leg with a given configuration are calculated. The reachable areas and CKMs of some CoGs obtained from different methods of moving the robot's body are shown.

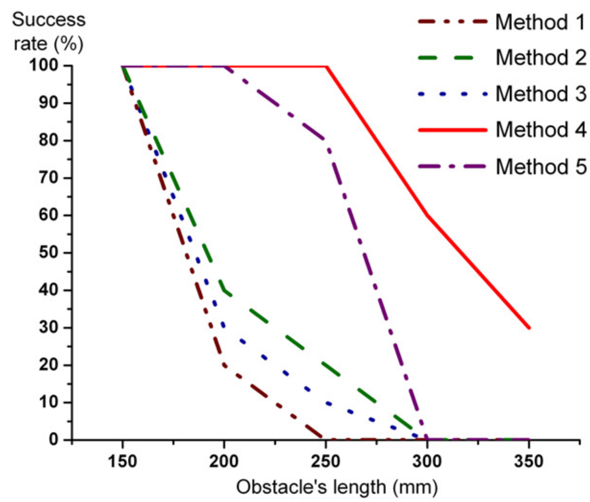


Fig. 19. Comparison of the traversability of the robot using different ways of moving the body.

7.1. Analysis

Since it is not able to replicate the full complexity of previous works, only the proposed method of moving the robot's body was compared to some previous *popular ways of moving body* to verify its advantages. The following methods of moving the robot's CoG were chosen in the comparison: the CoG is moved with shortest moving distance to the support polygon [21] (named as method 1), the CoG is moved to eliminate the four-leg supporting phases [19] (method 2), the CoG is moved to the most stable position [21,23, 24] (method 3), and the CoG is moved to maximize traversability as proposed in this paper (method 4). Besides, moving the robot's body with the proposed method without considering the desired set of foot positions was also compared (named as method 5). Again, note that the proposed algorithm is not compared to the algorithms of the cited reports, *only the proposed way of moving body* is compared to the ways of moving body that were *mentioned/used* in the abovementioned citations.

The reachable areas over the manageable area of the mentioned methods were calculated and shown in Fig. 17. The arcs-composed shape of the stable areas is similar with different foot positions and body's orientation. In this example, the robot was at 150 mm height, the feet were on a plane with the set of desired foot positions. The normal stroke of method 2 was 200 mm. The figure showed that exploiting body movement gave the robot highest reachability of the next swing leg, comparing to the other methods of moving the CoG. It brought the most forward acceptable terrain cells for the next swing leg along a desired moving direction. The result also showed that CKM clearly represents reachability of the next swing leg along a given direction. By observing the CKM, the robot can know its exact ability of overcoming the local terrain. Therefore, it helps in generating suitable decisions to overcome the terrain such as swinging distance, swinging direction, or alternating the swinging order.

A test was also performed to compare the traversability of the robot using the mentioned ways of moving the body. In the test, the mission of the robot was to go from a box to another box by placing its legs only on three bars as shown in Fig. 18. The distances from the bars were randomly generated in each test from 100 to 350 mm. There were five groups of environments depending on the upper boundary of the maximum distance as: 150, 200, 250, 300, and 350 mm. The success rates of the tests using the methods in each group were shown in Fig. 19 in which the x-axis represents the upper boundary of the maximum distance. Moving the CoG with the shortest distance to the support polygon showed lowest traversability since it had very limited CKM for the next swing leg as shown in Fig. 17. The result of moving the CoG to the most stable position was not impressive. The reason was that it reduced the reachability of the fore swing legs when it kept the body backwards to increase the stability. The result of method 2 was nice since it had high CKM for both hind and fore legs. However, the impressive improvement of the traversability was only recognized when body movement was exploited. The success rate was up to 30% when the distances from the bars were varied within 100–350 mm (note that a distance must be within 300–350 mm at least). It is a notable result since the effective leg's length is only 520 mm (note that the workspace of the MR4 robot is quite limited because the range of the first joint of the leg is from 0° to 90°) and the environment was

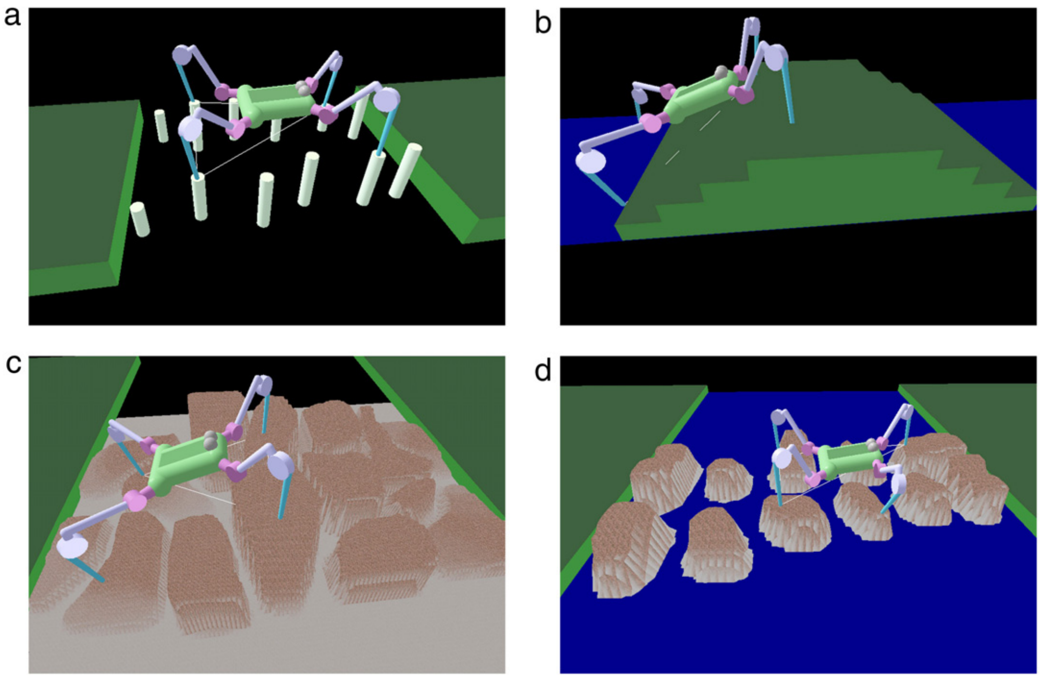


Fig. 20. Walking of MR4 robot using the proposed algorithm in some simulated challenging rough terrains.

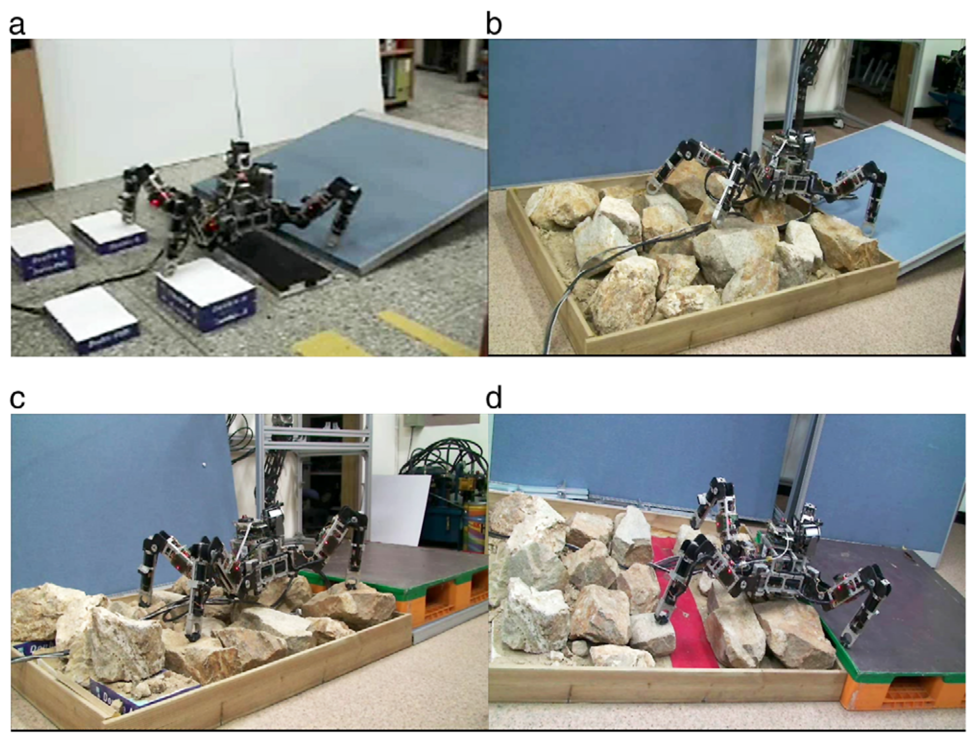


Fig. 21. MR4 walked over severely real rough terrains by using the proposed algorithm.

a really challenging one. Besides, the result of method 5 showed that evaluating the quality of a foot placement without caring the current configuration of the robot really reduced the robot's traversability.

7.2. Simulations and experiments

In this part, the proposed algorithm was validated in several types of rough terrains. In many cases where the next desired foothold is near the current foothold, maximizing the reachability of the next swing leg is unnecessary and may lead to some disadvantages such as backward motion, slow moving time. However, we did not eliminate that redundant action from the robot's behavior to emphasize the difference of the algorithm.

7.2.1. Simulation results

The proposed algorithm was validated in various popular rough terrains. Fig. 20(a) showed a snapshot when the robot was walking over some poles which were randomly placed along the path with different heights. Fig. 20(b) showed that the robot was capable of dealing with big steps. In Fig. 20(c), the robot walked over a typical rough terrain. Moreover, the algorithm was also validated in a rough terrain that was not conquered by previous robots as shown in Fig. 20(d). This rough terrain contained only some safe areas placed very far from each other. The distances between the rocks (the safe areas) were up to around 300 mm, that is nearly half the length of the robot's leg. Also, the rocks' height, size and shape were arbitrary as in the natural environment. The robot tried to keep its body low and in parallel with the local terrain to have a high traversability posture. In many cases, the robot must alternate the order of swinging the legs to have a high traversability posture for the next step. Since the method is purely geometric, the robot just needed 30–60 s for calculating the terrain cost map when the robot was set up. All other calculations were done in real-time with the main controller of MR4. In most of the tests, N_R was 30 and $\phi_{\max}$ was 15°.

7.3. Experimental results

Four experiments in real rough terrains were carried out to validate the proposed algorithm. In the experiments, the robot was fully autonomous except for the given map and walked in the environment without any intervention from the human. In the first experiment, the robot traversed a terrain composed of some office-items placed on the floor to make obstacles, ditches and a slope as shown in Fig. 21(a). The largest distance between two adjacent safe areas was up to 250 mm in length. The three remained environments were built from real rocks as shown in Fig. 21(b)–(d). The environment in Fig. 21(b) had a convex shape while the environment in Fig. 21(c) had a concave shape. Meanwhile, there was a large ditch in the fourth experiment as shown in Fig. 21(d). The robot needed to drastically adjust its body's position and orientation to adapt with the real terrains. Using the proposed algorithm, the robot finally overcame all the real challenging terrains. The robot showed the same behavior of maximizing the traversability in both tests as well as in the simulations.

8. Conclusion

In this paper, three major topics of quadruped walking robots in severely challenging terrains, such as, body movement, foothold selection, and gait selection were investigated. Using the proposed methods, the robot did have a high traversability in rough terrains. The result showed that the size of the obstacle that the robot can overcome was up to 70% of its leg's length. Also, the algorithm ran very fast in real-time since it is a purely geometric method. Besides, many useful concepts to study locomotion of legged robots in 3D rough terrains such as stable area, manageable area, and CKM were introduced. The theories and verification were presented with a quadruped robot that has a sprawl posture. However, the validation of the proposed method is reserved with upright posture quadruped robots.

Acknowledgments

This work was supported by Dual Use Projects of MKE (Ministry of Knowledge and Economy) contract titled "Development of Quadruped Legged Robot Platform Technology" and Korea Institute for Advancement in Technology (KIAT) through the Workforce Development Program in Strategic Technology. An earlier version of this document was approved by ADD Dual Use Technology Center for Public Release.

Appendix. Supplementary data

Supplementary material related to this article can be found online at doi:10.1016/j.robot.2011.08.007.

References

[1] R.B. McGhee, G.I. Iswandhi, Adaptive locomotion of a multi-legged robot over rough terrain, IEEE Transactions on Systems, Man, and Cybernetics SMC-9 (4) (1979) 176–182.
[2] R.B. McGhee, A.A. Frank, On the stability properties of quadruped creeping gaits, Mathematical Biosciences 3 (1968) 331–351.
[3] R. Playter, M. Buehler, M. Raibert, BigDog, in: Proceeding of 2006 SPIE, vol. 6230.
[4] H. Kimura, Y. Fukuoka, A.H. Cohen, Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts, International Journal of Robotics Research 26 (5) (2007) 475–490.
[5] S. Hirose, A study of design and control of a quadruped walking vehicle, International Journal of Robotics Research 3 (2) (1984) 113–133.
[6] D. Messuri, C. Klein, Automatic body regulation for maintaining stability of a legged vehicle during rough terrain locomotion, IEEE Journal of Robotics and Automation 1 (3) (1985) 132–141.
[7] T.T. Lee, G.L. Shih, A study of the gait control of a quadruped walking vehicle, IEEE Journal of Robotics and Automation 2 (2) (1986) 61–69.
[8] C.L. Shih, C.A. Klein, An adaptive gait for legged walking machines over rough terrain, IEEE Transactions on Systems, Man, and Cybernetics 23 (4) (1993) 1150–1155.
[9] S. Bai, K.H. Low, A new free gait generation for quadrupeds based on primary/secondary gait, in: Proceeding of 1999 IEEE Int. Conf. Robot. Autom., ICRA 1999, vol. 2, pp. 1371–1376.
[10] P.K. Pal, K. Jayarajan, Generation of free gait—a graph search approach, IEEE Transaction on Robotics and Automation 7 (3) (1991) 299–305.
[11] D.J. Pack, H.S. Kang, Free gait control for a quadruped walking robot, Laboratory Robotics and Automation 11 (2) (1999) 71–81.
[12] C. Eldershaw, M. Yim, Motion planning of legged vehicles in an unstructured environment, in: Proceeding of 2001 IEEE Int. Conf. Robot. Autom., ICRA 2001, vol. 4, pp. 3383–3389.
[13] T. Bretl, Motion planning of multi-limbed robots subject to equilibrium constraints: the free-climbing robot problem, International Journal of Robotics Research 25 (4) (2006) 317–342.
[14] J. Estremera, P.G. de Santos, Generating continuous free crab gaits for quadruped robots on irregular terrain, IEEE Transactions on Robotics 21 (6) (2005) 1067–1076.
[15] K. Hauser, T. Bretl, J.-C. Latombe, Motion planning for legged robots on varied terrain, International Journal of Robotics Research 27 (11–12) (2008) 1325–1349.
[16] A. Chillian, H. Hirschmuller, Stereo camera based navigation of mobile robots on rough terrain, in: Proceeding of 2009 IEEE Int. Conf. Robots Syst., IROS 2009, USA, October 11–15.
[17] B. Mitchell, A.G. Hofmann, B.C. Williams, Search-based foot placement for quadrupedal traversal of challenging terrain, in: Proceeding of 2007 IEEE Int. Conf. Robot. Autom., ICRA 2007, Italy, April 10–14, pp. 1461–1466.
[18] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, Learning, planning, and control for quadruped locomotion over challenging terrain, International Journal of Robotics Research 30 (2) (2011) 236–258.
[19] J. Zico Kolter, Andrew Y. Ng, The Stanford LittleDog: a learning and rapid replanning approach to quadruped locomotion, International Journal of Robotics Research 30 (2) (2011) 150–174.
[20] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, W. Burgard, A Bayesian regression approach to terrain mapping and an application to legged robot locomotion, Journal of Field Robotics 26 (10) (2009) 789–811.
[21] P.D. Neuhaus, J.E. Pratt, M.J. Johnson, Comprehensive summary of the Institute for Human and Machine Cognition's experience with LittleDog, International Journal of Robotics Research 30 (2) (2011) 216–235.
[22] D. Pongas, M. Mistry, S. Schaal, A robust quadruped walking gait for traversing rough terrain, in: Proceeding of 2007 IEEE Int. Conf. Robot. Autom., ICRA 2007, Italy, April 10–14, pp. 1461–1466.
[23] P. Vernaza, M. Likhachev, S. Bhattacharya, S. Chitta, A. Kushleyev, D.D. Lee, Search-based planning for a legged robot over rough terrain, in: Proceeding of 2009 IEEE Int. Conf. Robots Syst., IROS 2009, USA, October 11–15.
[24] H. Tsukagoshi, S. Hirose, K. Yoneda, Maneuvering operations of the quadruped walking robot on the slope, in: Proceeding of 2002 IEEE Int. Conf. Robots Syst., IROS 2002, vol. 2, pp. 863–869.
[25] W. Chen, K. Low, S. Yeo, Adaptive gait planning for multi-legged robots with an adjustment of center-of-gravity, Robotica 17 (1999) 391–403.
[26] E. Oks, M. Sharir, Minkowski sums of monotone and general simple polygons, Discrete and Computational Geometry 35 (2) (2006) 223–240.
[27] T.H. Kang, H.S. Kim, T.Y. Son, H.R. Choi, Design of quadruped walking and climbing robot, in: Proceeding of 2003 IEEE Int. Conf. Robots Syst., IROS 2003, pp. 619–624.

Vo-Gia Loc received the B.S. degree in Mechanical engineering from Ha Noi University of Technology in Vietnam, 2003, and the M.S. degree mechanical engineering from Sung Kyun Kwan University, Korea, in 2006, where he is currently working toward the Ph.D.degree. His research interests include legged locomotion, walking and climbing robot.

Ig Mo Koo received the B.S. degree in mechanical engineering from Myongji University, Yongin, Korea, in 2003, the M.S. degree in mechanical engineering from the Sung Kyun Kwan University, Suwon, Korea, in 2005, where he is currently working toward the Ph.D. degree in mechanical engineering from Sung Kyun Kwan University. His research interests include artificial muscle actuators, haptics, tactile display, biomimetics and quadruped walking robot systems.

Duc Trong Tran Trong received the B.S. degree in Mechatronics from Ho Chi Minh City University of Technology in Vietnam in 2005, where he is currently working toward the M.S. degree in mechanical engineering from Sung Kyun Kwan University. His research interests include biological inspired control and adaptive control of quadruped walking robot.

Sangdoek Park is currently with the Division of Applied Robot Technology, Korea Institute of Industrial Technology, Ansan, Korea.

Hyungpil Moon received the B.S. and M.S. degrees in mechanical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1996 and 1998, respectively, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2005. He was a postdoctoral fellow at Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. He is currently an assistant professor in the school of mechanical engineering, Sungkyunkwan University, Suwon, Korea. His research interests include distributed manipulation, localization and navigation of multi-agent systems, and biomimetic robotics.

Hyouk Ryeol Choi received the B.S. degree from Seoul National University, Seoul, Korea, in 1984, the M.S. degree from the Korea Advanced Technology of Science and Technology (KAIST), Taejon, Korea, in 1986, and the Ph.D. degree from the Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1994. Since 1995, he has been with Sungkyunkwan University, Suwon, Korea, where he is currently a Professor in the School of Mechanical Engineering. He was an Associate Engineer with LG Electronics Central Research Laboratory, Seoul, Korea, from 1986 to 1989. From 1993 to 1995, he was with Kyoto University, Kyoto, Japan, as a grantee of scholarship funds from the Japanese Educational ministry. He visited the Advanced Institute of Industrial Science Technology (AIST), Tsukuba, Japan, as a JSPS Fellow from 1999 to 2000. He was a visiting scholar in Biorobotics Laboratory in the University of Washington from 2008 to 2009. He served as an Associate Editor in IEEE Transactions on Robotics. He is now an Editor of International Journal of Control, Automation and Systems (IJCAS), and Associate Editor of Journal of Intelligent Service Robotics. His interests include dexterous mechanisms, field application of robots, and artificial muscle actuators.