UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

2004 route des Lucioles
B.P. 93
06902 Sophia-Antipolis
France

Rapports de Recherche

N°1767

Programme 4

Robotique, Image et Vision

MOTION PLANNING OF LEGGED ROBOTS: THE SPIDER ROBOT PROBLEM

Jean-Daniel BOISSONNAT
Olivier DEVILLERS
Franco P. PREPARATA
Leonbattista DONATI

Octobre 1992

Motion Planning of Legged Robots: the Spider Robot Problem*

Planification de trajectoires de robots à pattes : le problème du robot araignée.

Jean-Daniel Boissonnat[†] **Olivier Devillers**[†]
Franco P. Preparata[‡] **Leonbattista Donati**[†]

[†]INRIA, 2004 Route des Lucioles, B.P.109, 06561 Valbonne cedex (France), Phone : +33 93 65 77 60, E-mail : `firstname.name@sophia.inria.fr`. *Work of these authors has been supported in part by the ESPRIT Basic Research Actions Nr. 3075 (ALCOM), 7141 (ALCOMII) and 6546 (PROMOtion).*

[‡]Department of Computer Science, Brown University, Providence, RI 02912, (USA), E-mail : `franco@cs.brown.edu`. *Work of this author has been supported in part by NSF Grant CCR-91-96176.*

Abstract

We consider the problem of planning motions of a simple legged robot called the spider robot. The robot is modelled as a point where all its legs are attached, and the footholds where the robot can securely place its feet consist of a set of n points in the plane. We show that the space $\mathcal{F}$ of admissible and stable placements of such robots has size $\Theta(n^2)$ and can be constructed in $O(n^2 \log n)$ time and $O(n^2)$ space. Once $\mathcal{F}$ has been constructed, we can efficiently solve several problems related to motion planning.

Keywords: Spider robot, Legged robot, Motion planning

Résumé

Nous considérons le problème de la planification de trajectoires pour un robot à pattes simple que nous appelons le robot araignée. Le robot est modélisé par un point où toutes les pattes sont accrochées, et les points d'appui où le robot peut poser ses pattes en toute sécurité consistent en un ensemble de n points du plan. Nous montrons que l'espace $\mathcal{F}$ des placements admissibles et stables pour un tel robot a une taille $\Theta(n^2)$ et peut être construit en temps $O(n^2 \log n)$ en utilisant $O(n^2)$ mémoire. $\mathcal{F}$ étant construit, on peut résoudre efficacement certains problèmes de planification de trajectoires.

Mots-clés : Robot araignée, robot à pattes, planification de trajectoires.

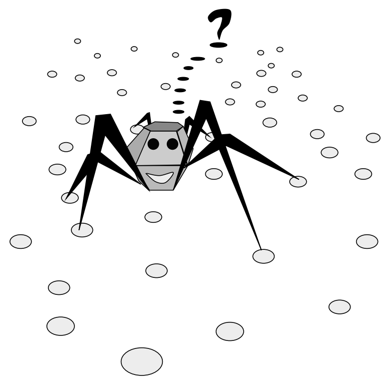

Figure 1: The spider robot problem

1 Introduction

Although legged robots already exist, until now researchers have been more interested in their dynamics and their control (see general references[RR90, RR84]) than in motion planning problems : the literature on the topic is almost nonexistent and we are aware of only a very few papers that consider the problem of planning the mouvement of the legs of a legged robot moving amidst obstacles and dangerous areas[HNKU84, HK91].

The robot we consider here is a simple legged robot, called a spider robot. The body is a single point; a number of segments (the legs), whose lengths may vary within bounds, are attached to the body (see Figure 1). This model has been inspired by the Ambler,[BHK+89] a legged robot developped at Carnegie Mellon University.

The constraints on the robot motion are of two types; first, the robot feet have to rest on reachable footholds (feasibility constraint) and, second, each position of the robot must be stable (stability constraint). In this paper, the footholds are points in the plane and we show that the space of valid configurations, that is the set of all the positions of the robot body for which there exists a feasible and stable placement of its feet, has size $\Theta(n^2)$ where n is the number of predefined footholds. Next we give an algorithm that computes this subset of the plane in time $O(|\mathcal{A}|\log n)$ and $O(|\mathcal{A}|)$ space. $|\mathcal{A}|$ is the size of a certain arrangement of n circles, which in the worst-case is $\Omega(n^2)$, but in most realistic situations will be $O(n)$. This set of stable configurations can readily be used to decide whether or not there exists a feasible and stable path between two given placements. It can also be used in combination with $\mathcal{A}$ to compute a sequence of stable placements for the feet along that path.

The paper is organized as follows: in the next section, we introduce the necessary definitions for the robot and we make precise the problems to be solved. In Section 3, we study the space of feasible and stable configurations and give a tight bound on its complexity. Section 4 presents the algorithm that computes the space of stable configurations. Applications of the present work to solve several motion planning problems and some concluding remarks are mentionned in the last two sections.

2 Definitions

The body of a spider robot is a single point and will be denoted G; each foot of the robot can reach the points of the plane inside the disk of radius R centered at G: this disk is called the *range of action* of the robot. If q is a point in the plane; $D(q)$ denotes the closed disk of radius R centered at q.

The footholds where the robot feet can stay safely consist of a set $\mathcal{M}$ of n points (the *sites*) of the Euclidean plane.

As the problem we consider is essentially planar, all figures, except Figure 1, will be drawn in the plane and we will assume that G is also a point of the plane (we identify G and its vertical projection onto the plane).

2.1 Configurations and Placements

We call *configuration* of the robot a position of its body G. For a given configuration, we call *placement* of the robot a set of pairings between the robot feet and some points of $\mathcal{M}$. A placement is defined by $\mathcal{I}$, the set of resting legs and by $p_i \in \mathcal{M}, i \in \mathcal{I}$, the position of the feet. We will say that a placement is an *l-leg placement* if $|\mathcal{I}| = l$.

For a given configuration, a placement is said *l-feasible* if there is a set of indices $\mathcal{I}$ such that

$$\forall i \in \mathcal{I}, d(G, p_i) \leq R \text{ and } |\mathcal{I}| \geq l \tag{1}$$

where $d(A, B)$ is the Euclidean distance between two points of the plane. Equation (1) says that there must be at least l sites in the range of action of the robot.

A configuration is said to be *l-feasible* if there is an l-feasible placement for this configuration. In Figure 2, configuration 1 is 3-feasible and configuration 2 is 4-feasible (the position of G is given by the black square).

Lastly, for a given configuration, a placement is said to be *stable* if the following condition holds:

$$G \in \mathrm{CH}(\{p_i, i \in \mathcal{I}\}), \tag{2}$$

where $\mathrm{CH}(S)$ denotes the convex hull of S. As above, a robot configuration is *stable* if there is a stable placement for the configuration (only configuration 1 is stable in Figure 2).

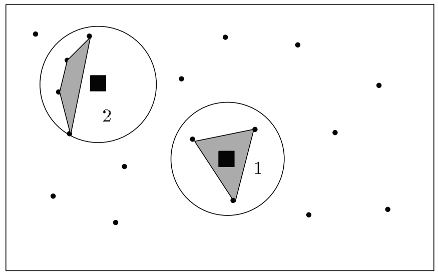

Figure 2: Two configurations

2.2 Paths and Motions

We call *l-path* of the spider robot, a continuous trajectory $\mathcal{C}$ of l-feasible configurations; that is, for each position of G along $\mathcal{C}$ there must be a l-feasible placement. A *l-motion* consists of a l-path $\mathcal{C}$ of G together with a sequence of possible placements.

A *stable path* or *motion* is a path or a motion in which every position of the robot body satisfies the stability condition 2.

It should be added that along an l-motion there could be configurations where the robot has to change its placement: that is, without changing the position of its body, it has to change the resting places of its legs. This is possible if the robot has an $l+1$th leg; with this additional leg, it can change its resting sites while always keeping l legs on the ground.

3 Feasible and Stable Configurations

3.1 Relaxing One Constraint

In this section, we show that the problem can be easily solved if only one constraint, the feasibility or the stability constraint, is considered.

Clearly, if we relax the feasibility constraint, the set of stable configurations is the convex hull of the sites, which can be computed in $O(n \log n)$ time.[PS85]

On the other hand, if we relax the stability constraint, the set of l-feasible (not necessarily stable) configurations can be deduced from the order l Voronoi diagram. Indeed, for a given configuration G of the robot, there exists an l-feasible placement as soon as the placement defined by setting l feet on the l nearest footholds is l-feasible.

More precisely, we compute the superposition of the order l and order $l-1$ Voronoi diagram. For any point of a given cell V_i of this diagram, the l nearest footholds are the same, say $p_{i_1}, p_{i_2}, \ldots, p_{i_l}$ and the furthest among these ones

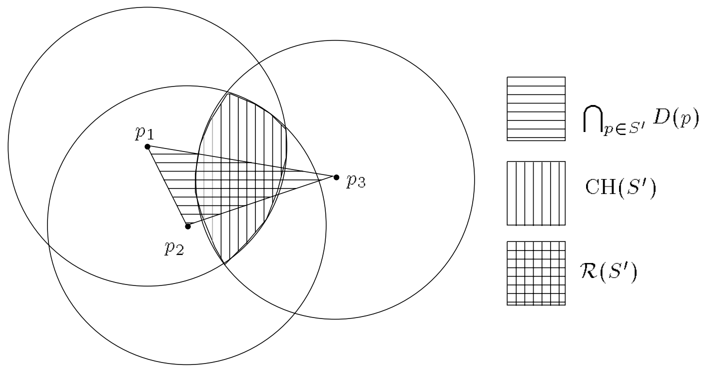

Figure 3: Stability region defined by $S = \{p_1, p_2, p_3\}$

is a unique site p_{i_l}. Thus $W_i = V_i \cap D(p_{i_l})$ is the portion of V_i consisting of l-feasible configurations (all footholds can be reached as soon as p_{i_l} is reachable since it remains the furthest foothold inside V_i). The union W of the W_i for all the cells V_i of the diagram is the whole set of l-feasible configurations. The sizes of the order $l-1$ and l Voronoi diagrams, as well as the size of their superposition are $O(nl)$ and they can be computed in $O(n \log n + nl^2)$ time.[Lee82, AGSS87] Constructing W can be done within the same time bound.

Unfortunatly, the Voronoi diagram does not solve the problem of the stability of the robot. It may happen that the nearest footholds placement is not stable while there exists another placement which is both feasible and stable.

The rest of the section and of the paper will be concerned with both constraints.

3.2 Preliminary Results

The set $\{D(p), p \in \mathcal{M}\}$ defines an arrangement $\mathcal{A}$ of circles.

The number of edges of $\mathcal{A}$ will be called the size of $\mathcal{A}$ and will be denoted by $|\mathcal{A}|$. By slightly adapting standard techniques for line segments, $\mathcal{A}$ can be computed in $O(|\mathcal{A}| \log n)$ time.[PS85]

For any $S \subseteq \mathcal{M}$, $|S| \geq 3$, we call *stability region* defined by S the following subset of the Euclidean plane:

$$\mathcal{R}(S) = \mathrm{CH}(S) \cap (\bigcap_{p \in S} D(p)). \tag{3}$$

Figure 3 shows the stability region defined by a subset of three sites.

Definition 1 *$\mathcal{F}$ denotes the union of the stability regions associated with all the subsets of $\mathcal{M}$ of cardinality 3:*

$$\mathcal{F} = \bigcup_{\substack{T \subseteq \mathcal{M} \\ |T|=3}} \mathcal{R}(T). \tag{4}$$

$\mathcal{F}$ is the set of stable 3-feasible configurations.

Lemma 2 *For any $S \subseteq \mathcal{M}$, with $|S| \geq 3$, we have*

$$\mathcal{R}(S) \subseteq \mathcal{F}. \tag{5}$$

Proof. As is well known,

$$CH(S) = \bigcup_{\substack{T \subseteq S \\ |T|=3}} CH(T).$$

Thus

$$\begin{aligned}
\mathcal{R}(S) &= \left(\bigcup_{T \subseteq S\ ,\ |T|=3} CH(T) \right) \cap \left(\bigcap_{p \in S} D(p) \right) \\
&= \bigcup_{T \subseteq S\ ,\ |T|=3} \left(\mathcal{R}(T) \cap \left(\bigcap_{p \in S \setminus T} D(p) \right) \right) \\
&\subseteq \bigcup_{T \subseteq S\ ,\ |T|=3} \mathcal{R}(T) \subseteq \mathcal{F}
\end{aligned}$$

$\square$.

The lemma shows that the complete information about stable regions is contained in $\mathcal{F}$. In particular, if there is no stable 3-leg placement, there is no stable l-leg placement with $l > 3$.

This is the reason why we shall focus our attention on set $\mathcal{F}$ and take $l = 3$ in the sequel. However, this is not crucial and our analysis could be extended in a straightforward way to the analysis and the construction of the set of stable l-feasible configurations ($l > 3$).

$$\mathcal{F}_l = \bigcup_{T \subseteq S\ ,\ |T|=l} \mathcal{R}(T). \tag{6}$$

In particular, Lemma 2 can be generalized and we can prove that for any $S \subseteq \mathcal{M}$, with $|S| \geq l$, we have $\mathcal{R}(S) \subseteq \mathcal{F}_l$.

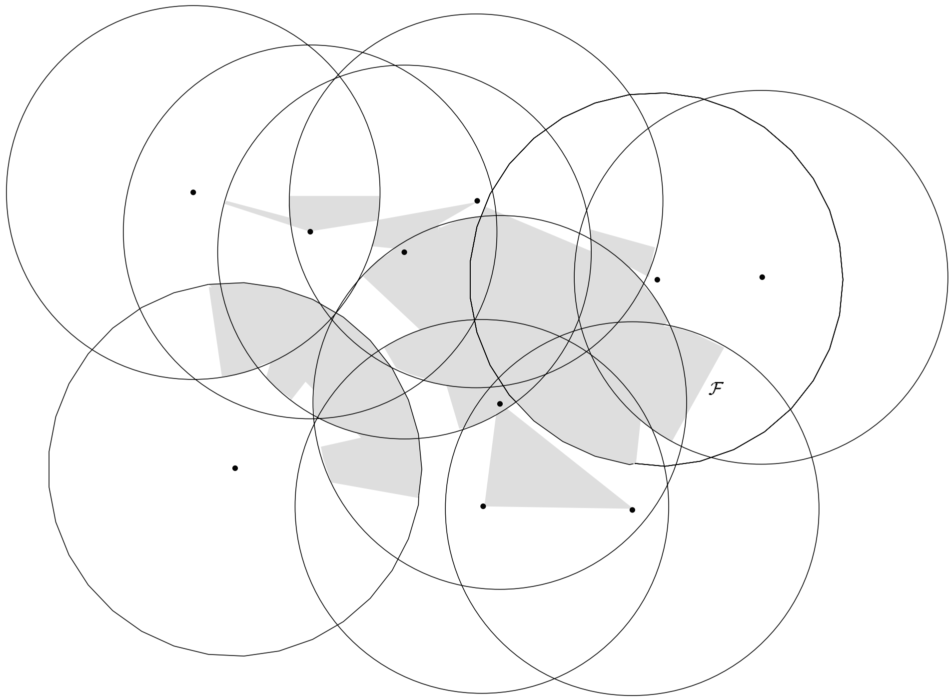

Figure 4: An example of free space of a spider robot

The boundary of $\mathcal{F}$, denoted by $\delta(\mathcal{F})$, consists of a disjoint union of cycles whose edges are arcs of circles $C_i = \delta(D(p_i))$ and straight-line segments belonging to lines passing through two sites (see Figure 4).

We shall classify the vertices of $\delta(\mathcal{F})$ on the basis of their internal angle (*convex* if the internal angle is less than π, *concave* otherwise) and on the basis of their incident edges (SS for segment-segment, CC for circular arc-circular-arc, and SC for segment-circular arc).

It follows from Equation (3) that, for any point x of the interior of a cell , of $\mathcal{A}$, the set $\mathcal{M} \cap D(x)$ of sites inside $D(x)$ is invariant. This set will be denoted S_Γ and $\mathcal{R}(S_\Gamma)$ will be denoted $\mathcal{R}($,). We denote $\mathcal{A}_l$ the set of cells , of $\mathcal{A}$ such that $|S_\Gamma| \geq l$.

Lemma 3 *Let , be a cell of* $\mathcal{A}_3$*. We have*

$$\mathcal{F} \cap , = \mathcal{R}(,) \cap , .$$

Proof. From Lemma 2, we have

$$\mathcal{R}(,\ ) \cap ,\ \subseteq \mathcal{F} \cap ,\ .$$

Conversely, assume that $x \in \mathcal{F} \cap ,$. Then, by definition, there exist three sites p_1, p_2, and p_3, such that

$$x \in \mathcal{R}(\{p_1, p_2, p_3\}) \cap ,$$

As the disks $D(p_i)$ $(i = 1, 2, 3)$ cannot intersect the boundary of , transversally,

$$\mathcal{R}(\{p_1, p_2, p_3\}) \subseteq \mathcal{R}(,\ ),$$

which proves the lemma. □.

Corollary 4 *Each SS vertex of $\delta(\mathcal{F})$ is a site $p \in \mathcal{M}$.*

Proof. Let u be an SS vertex of $\delta(\mathcal{F})$, intersection of two straight-line edges. If u belongs to region , of $\mathcal{A}$, then, by Lemma 3, u is a vertex of $\mathcal{R}(,\ )$ and by Equation (3), u must be a vertex of $\mathrm{CH}(S_\Gamma)$ that is a site in , $\subseteq \mathcal{M}$. We can also remark that every SS vertex is convex. □.

Corollary 5 *The vertices of $\delta(\mathcal{F})$ either are sites or belong to the boundary of a cell of the arrangement $\mathcal{A}$.*

Proof. We already know that SS vertices are sites. On the other end, if a vertex is not an SS vertex, then it is defined by at least one arc of circle and thus belongs to the boundary of a cell of $\mathcal{A}$. □.

3.3 Complexity of $\delta(\mathcal{F})$

Lemma 6 *Any edge of the arrangement $\mathcal{A}$ contains at most four SC vertices.*

Proof. Let circular arc C be an edge of $\mathcal{A}$ contained in $m \geq 3$ disks and let q be the center of the circle containing C. There is a unique maximal subset $S \subseteq \mathcal{M}$ with $q \in S$ and $|S| = m$ such that

$$C \subset \bigcap_{p \in S} D(p). \tag{7}$$

Let $P = \mathrm{CH}(S)$. If we define $S' = S - \{q\}$ and $P' = \mathrm{CH}(S')$, Equation 7 implies that

$$C \subset \bigcap_{p \in S'} D(p) \tag{8}$$

and thus, if $m > 3$,

$$C \cap P' \subset \mathcal{R}(S') \subset \mathcal{F}. \tag{9}$$

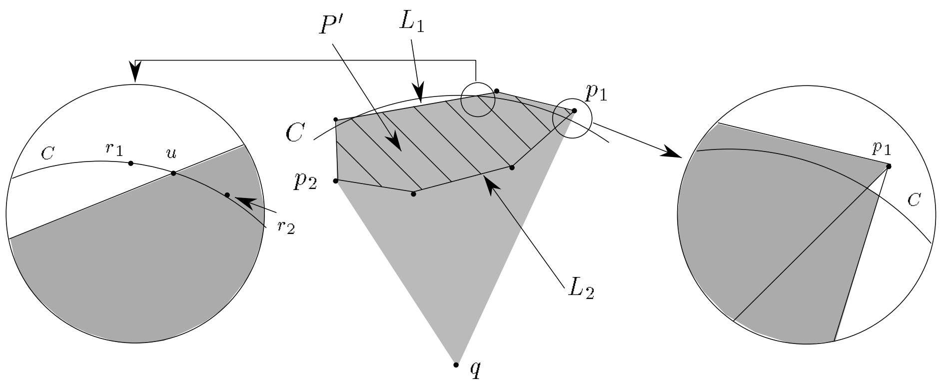

Figure 5: Mixed vertices on C

Let p_1 and p_2 be the two points of P' belonging to the two lines passing through q and tangent to P'. Then, if T is the triangle q, p_1, p_2, we have

$$C \cap T \subset \mathcal{R}(\{q, p_1, p_2\}). \tag{10}$$

From Equations 9 and 10, we deduce that an SC vertex on C is either an intersection between C and P' or an intersection between C and T (specifically, an intersection between C and either of the supporting segments joining q to P').

We consider first the case of an intersection between C and P'. Let L_1 be the polygonal chain consisting of all the common edges of P and P' (p_1 and p_2 being its two end points). We claim that, if $|S'| \geq 3$, none of the intersections of L_1 and C can be an SC vertex of $\delta(\mathcal{F})$. Indeed, let u be one of those intersections and consider a point r on C sufficiently close to u :

- if $r \notin P$, (r_1 in Figure 5) it cannot correspond to a stable placement of the robot since the only sites of $\mathcal{M}$ which are in $D(r)$ are contained in S and r is external to the convex hull of S, thus $r \notin \mathcal{F}$;
- if $r \in P$, (r_2 in Figure 5) then r is a stable configuration and there exists a placement that does not involve q since P' has at least three vertices. Thus r is in the interior of $\mathcal{F}$ and C cannot be an edge of $\mathcal{F}$ in the neighborhood of u.

Let us now consider the polygonal chain of P', called L_2 in Figure 5. Its intersections with C can actually be SC vertices but simple considerations of convexity show that there are at most two such vertices. As for the supporting segments qp_1 and qp_2, each can contribute at most one SC vertex, there is at most four SC vertices of $\delta(\mathcal{F})$ on C.

In the case where $|S'| = 2$ then $C \cap T \neq \emptyset$ and its portions belong to $\delta(\mathcal{F})$. This intersection is composed of one or two circular arcs and gives two or four SC vertices.

In each case, we can conclude that C contains at most four SC vertices. $\square$.

Now that we have bounded the number of mixed vertices lying on each arc of $\mathcal{A}$, we can state the following theorem where $|\mathcal{A}|$ denotes the size of $\mathcal{A}$.

Theorem 7 *The geometric complexity of $\delta(\mathcal{F})$ is $\Theta(|\mathcal{A}|)$.*

Proof. The vertices of $\delta(\mathcal{F})$ are sites ($O(n)$), or vertices of $\mathcal{A}$ ($O(|\mathcal{A}|)$), or SC vertices ($O(|\mathcal{A}|)$ from Lemma 6). Since, $\delta(\mathcal{F})$contains as many edges as vertices, the same bound $O(|\mathcal{A}|)$ holds for the number of edges of $\delta(\mathcal{F})$.

This bound is tight as is shown by the following example. $\square$.

3.4 Worst-case Size Example

In the worst case, the complexity of $\mathcal{A}$ is $\Omega(n^2)$. We exhibit in this section an example of a configuration set $\mathcal{F}$ of size $\Omega(n^2)$.

Put $\frac{n}{2}$ sites at the vertices of a regular polygon P' and $\frac{n}{2}$ other sites $p_1, \dots, p_{\frac{n}{2}}$ at the vertices of a larger regular polygon P with the same center as the first one. (Figure 6a. It is possible to choose the diameters of the polygons such that

- the size of the portion of $\mathcal{F}$ called $\mathcal{F}_i$, contained in $D(p_i)$ is $\Omega(n)$ for $i = 1, \dots, \frac{n}{2}$;
- the $\mathcal{F}_i$ are disjoint.

This last condition is ensured by the following construction (refer to Figure 6b. If 2ε is the diameter of P' and p_1 is at a distance $2R - \varepsilon$ from the center O of P', $\mathcal{F}_1$ has a diameter smaller than 2ε and crosses the circle C of center O and radius R. $\mathcal{F}_1$ does not intersect $\mathcal{F}_2$ if the disk $D(p_2)$ is disjoint from $\mathcal{F}_1$. This is possible by placing p_2 at a distance $O(\varepsilon)$ from p_1, as shown in Figure 6b. By repeating this construction, we obtain a set of stable configurations of size $\Omega(n^2)$.

In this example, $\mathcal{F}$ has $\Omega(n)$ connected components, each of size $\Omega(n)$ but it is easy to slightly deform the figure so as to connect all the $\mathcal{F}_i$ and obtain a connected component of $\mathcal{F}$ of size $\Omega(n^2)$. Thus, in the worst-case, a single component of $\mathcal{F}$ may have size $\Omega(n^2)$.

This example can also be adapted to reach the bound $O(|\mathcal{A}|)$ of Theorem 7 for $|\mathcal{A}| = o(n^2)$. For example by choosing $|P| = O(1 + \frac{|\mathcal{A}|}{n})$ and $|P'| = O(n)$.

However, these worst case examples will not be encountered in most practical situations and much better bounds are to be expected. In order to illustrate

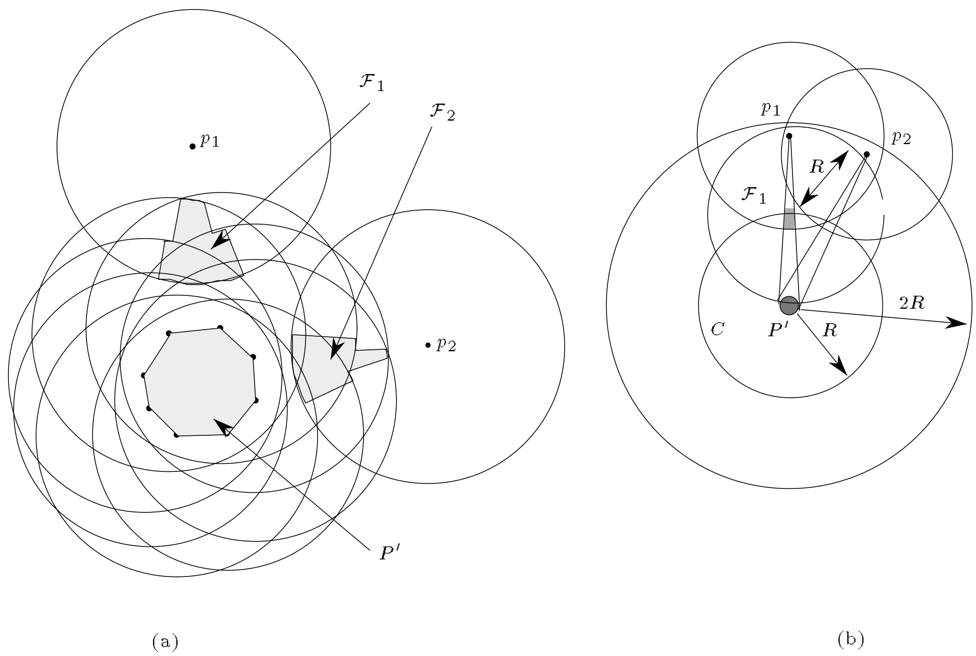

Figure 6: Construction of $O(n^2)$ configuration space

this point, let k be the maximum number of disks of $\{D(p), p \in \mathcal{M}\}$ that can cover a point of the plane. k will be called the *density* of the footholds. Clearly k is not larger than n and in case of *sparse footholds*, it is a constant that does not depend on n. It can be shown that $|\mathcal{A}| = O(kn)$ [Sha91]. Thus, in case of sparse footholds, the sizes of $\mathcal{A}$ and $\mathcal{F}$ are linearly related to the number of footholds.

4 Computation of $\delta(\mathcal{F})$

Our algorithm is based on Lemma 3.

First, we construct the arrangement $\mathcal{A}$ of the circles slightly adapting a standard $O(|\mathcal{A}| \log n)$ time algorithm.[PS85]

We then have to intersect each cell with its *stability hull*, i.e. the convex hull of the centers of the disks covering the cell. Note that we do not need to recompute the entire hull for every cell since the sets of covering disks of two adjacent cells differ only by one element. To exploit this property, we have to use a convex hull algorithm that can handle insertions and deletions. As in fact

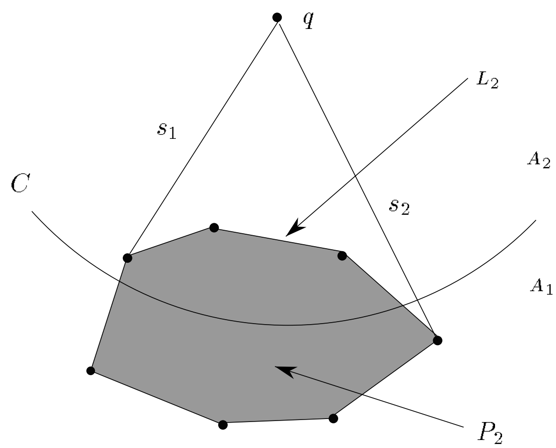

Figure 7: Construction of $\mathcal{F}$

everything is fixed and known from the beginning, we will use the algorithm for offline maintenance of convex hulls of Hershberger and Suri.[HS91]

Suppose we are in a cell A_1 of the arrangement and that we know, from the convex hull maintenance algorithm, the stability hull P_1 of A_1. Let A_2 be a cell adjacent to A_1, so that their common arc C turns its concavity towards A_2 (see Figure 7) and let q be the center of the circle containing C. The stability hull P_2 of A_2, which is $CH(P_1 \cup \{q\})$, is computed in $O(\log n)$ time using the convex hull maintenance algorithm which also provides the following informations:

- the polygonal chain L_2 whose edges are edges of P_1 and not of P_2,
- the two suporting lines s_1 and s_2 from q to P_1.

The situation shown in Figure 7 is similar to the one in Figure 5: as before the SC vertices belonging to C can only lie on L_i, s_1 and s_2. Here L_i denotes L_1 if L_1 is a line segment and L_2 otherwise. The search on s_1 and s_2 can be done in constant time; on the other hand, since C and L_i intersect at most twice, by binary search, we can locate the intersections between C and L_i in time $O(\log n)$.

Once we know, for a given cell, the SC vertices lying on its boundary and its stability hull, we can easily paste these informations to get the portion of $\delta(\mathcal{F})$ in the cell. Repeating this construction during the traversal of $\mathcal{A}$ gives the entire $\delta(\mathcal{F})$.

Theorem 8 *The boundary of $\mathcal{F}$ can be computed in time $O(|\mathcal{A}| \log n)$ and space $O(|\mathcal{A}|)$.*

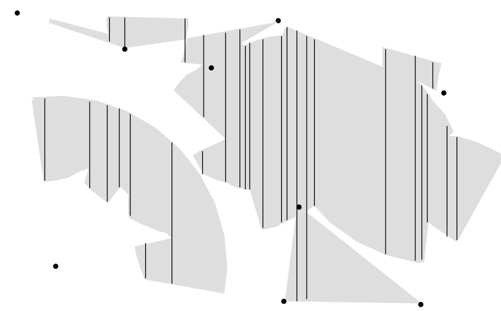

Figure 8: The trapezoidal decomposition of $\mathcal{F}$

Proof. For each cell of $\mathcal{A}$ we need $O(\log n)$ time to update the convex hull and $O(\log n)$ to compute the SC vertices on each arc. The construction of $\mathcal{F}$ from these items can then be done in time proportional to the size of $\mathcal{F}$, which is $O(|\mathcal{A}|)$. $\square$.

Furthermore, $\mathcal{F}$ can be decomposed into trapezoids with no overcost : trapezoids are obtained by extending from each vertex of $\mathcal{F}$ a vertical line and keeping the portion that is included in $\mathcal{F}$ and contains the vertex. See Figure 8 for the trapezoidal decomposition of the example of Figure 4.

5 Applications

5.1 Placements

Question 1 : *For a given configuration, does there exist a feasible and stable placement ?*

To answer the query, we have to decide whether or not G lies in $\mathcal{F}$. The problem reduces to locating G in the planar map $\mathcal{F}$. As a location query in a planar map of size m can be done in time $O(\log m)$ after a preprocessing that requires $O(m \log m)$ time and $O(m)$ space,[PS85] we have

Theorem 9 *There exists a data structure such that, for a given position of the body G of the robot, we can decide if there exists a stable placement in time $O(\log n)$. The data structure requires $O(|\mathcal{F}|)$ space and can be constructed in $O(|\mathcal{F}| \log n)$ additional time, once $\mathcal{F}$ has been computed.*

Question 2 : *For a given feasible and stable configuration G, find a placement.*

We first determine the cell , of $\mathcal{A}$ which contains G. Let S_Γ be the set of sites whose disks cover , , $CH(S_\Gamma)$ their convex hull and x any point of S_Γ. We determine the edge yz of $CH(S_\Gamma)$ which is intersected by the ray issued from x and passing through G. xyz is a feasible and stable placement.

In order to construct $CH(S_\Gamma)$, we use a data structure proposed by Hershberger and Suri[HS91] that allows to search in history. This data structure, which is similar to the one used for constructing $\mathcal{F}$, stores in an implicit way all the $CH(S_\Gamma)$ for the different cells , of $\mathcal{A}$. The data structure can be constructed in $O(|\mathcal{A}| \log n)$ time and uses $O(|\mathcal{A}| \log n)$ space. Constructing $CH(S_\Gamma)$ and finding edge yz can then be done in $O(\log n)$ time.

Theorem 10 *There exists a data structure such that, for a given feasible and stable configuration of the robot, we can compute a stable placement in time $O(\log n)$. The data structure requires $O(|\mathcal{A}| \log n)$ space and can be constructed in $O(|\mathcal{A}| \log n)$ time.*

5.2 Paths

Question 3 : *For two given stable configurations, does there exist a stable path joining them ?*

To answer the query, we have to decide whether or not the two configurations belong to the same connected component of $\mathcal{F}$. Plainly, this can be done within the same bounds as for Question 1.

If one wants to construct the path, it suffices to look for a path inside $\mathcal{F}$, which can be done in time $O(|\mathcal{F}| \log n)$ and space $O(|\mathcal{F}|)$.

More precisely, we search a path in the adjacency graph of the trapezoidal map of $\mathcal{F}$ using breadth first search. This yields a sequence of adjacent trapezoids of $\mathcal{F}$, such that the first one contains S, the last one contains A and each trapezoid of $\mathcal{F}$ appears at most once in the sequence.

Theorem 11 *There exists a data structure such that, given two configurations S and A, we can decide if there exits a stable path joining S and A in time $O(\log n)$. The data structure requires $O(|\mathcal{F}|)$ space and can be constructed in $O(|\mathcal{F}| \log n)$ additional time, once $\mathcal{F}$ has been computed. Constructing such a path can be done within the same bounds as the preprocessing.*

It remains to find the sequence of placements along that path. This will be considered in the next section. We will need the additional result

Proposition 12 *Any stable path followed by the body of the robot intersects the trapezoids of $\mathcal{A}$ $O(k|\mathcal{A}|)$ times, if k is the density of the footholds.*

Proof. Each trapezoid of $\mathcal{A}$ that is covered by k disks is decomposed into $O(k)$ subcells by the convex hull of the k centers of the disks and thus into $O(k)$ trapezoids of $\mathcal{F}$. $\square$.

Question 4 : *For two given configurations belonging to the same connected component of $\mathcal{F}$, find the shortest stable path joining them ?*

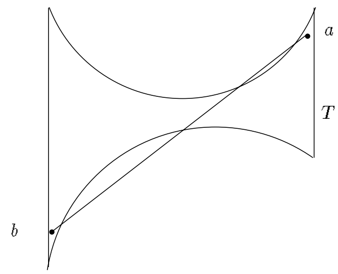

Figure 9: Crossing of a trapezoid

One may be interested in computing a shortest path for the body of the robot. We observe that a shortest path $\mathcal{C}$ is a succession of straight line-segments: indeed, the portions of $\mathcal{C}$ inside $\mathcal{F}$ are obviously line segments and, since each curved edge of $\delta(\mathcal{F})$ is inner convex, only straight edges of $\delta(\mathcal{F})$ can appear as parts of $\mathcal{C}$. As a consequence of this observation, we can compute a shortest path inside $\mathcal{F}$ by means of any known algorithm for computing a shortest path inside a polygonal region, for example the ones based on the visibility graph. By the result of Welzl,[Wel85] $\mathcal{C}$ can thus be computed in $O(|\mathcal{F}|^2)$ time.

Theorem 13 *There exists a data structure such that, given two configurations S and A belonging to the same connected component of $\mathcal{F}$, we can compute the shortest stable path joining S and A in time $O(|\mathcal{F}|^2)$.*

Before we give the algorithm for the actual computation of the motion, we show that a shortest path $\mathcal{C} \subset \mathcal{F}$, between two stable and feasible configurations S and A, is a stable path with the property that the path intersects only $O(|\mathcal{A}|)$ trapezoids of $\mathcal{A}$. (this result is to be compared with Proposition 12).

Proposition 14 *The intersection of a shortest path with any trapezoid of $\mathcal{A}$ consists of at most three line segments.*

Proof. Recall that each trapezoid of the trapezoidal decomposition of $\mathcal{A}$ is limited above and below by an arc of circle and on the left and on the right sides by vertical line segments (Figure 9).

We will show that given two points a and b of C belonging to the same trapezoid T of $\mathcal{A}$ the portion of a short path between a and b is the line segment ab. This, in particular, holds when a is the point where C enters for the first time T and b is the point where C exits T for the last time. The claim then follows from the fact that the intersection of a segment and a trapezoid consists of at most three segments (see Figure 9).

Let P be the convex hull of the centers of the disks covering T and let D be the intersection of the disks covering T. Since P and D are convex, $ab \subset P$ and $ab \subset T$. Thus $ab \subset \mathcal{F}$. □.

5.3 Motions

Question 5 : *Once a stable (possibly shortest) path $\mathcal{C}$ has been found, find a corresponding motion.*

The motion planning problem is solved by computing the sequence of leg placements as the robot body G moves along $\mathcal{C}$ in the trapezoidal map of $\mathcal{A}$. As four legs are always sufficient to find a stable motion if one exists, we assume in this paragraph that the robot has four legs.

We call *critical* placement a placement such that G cannot go further along $\mathcal{C}$ with the same footholds. There are two types of critical placements. A critical placement of *Type 1* occurs when the stability condition does not hold any more because G crosses an edge of the current stability polygon (i.e. the convex hull of the footholds actually used by the robot). The second type occurs when one of the resting legs cannot extend further because G is on a circle whose center is an actual foothold *Type 2*.

Let G be a critical configuration of $\mathcal{C}$ and let P be the convex hull of the sites contained in the disk $D(G)$. We show how to pass the two types of critical placement.

The first type of critical placement is the simplest to deal with: we have to put the free leg on a site ahead, in the direction of $\mathcal{C}$ and to lift a back leg (i.e. a leg whose foot is not on the edge p_2p_3 where G is : p_1 in Figure 10). More precisely, we compute (in time $O(\log n)$) the intersection of P and $\mathcal{C}$ and from this point we find the first vertices of P in counterclockwise (p_4 in Figure 10) and clockwise order (p_5). We assume here that p_4 and p_5 are distinct from p_2 and p_3 ; the other cases can be easily deduced from the discussion below. While keeping G at the same point, the free leg is put on p_5, the leg on p_1 is lifted and the robot body can then go on along $\mathcal{C}$ in the triangle p_2, p_3, p_5 until a new critical placement occurs. If the robot reaches the boundary of the triangle p_2, p_3, p_5, it can make similar leg moves : now the free leg is moved to p_4, the back leg in p_3 is lifted and the motion is continued in the triangle p_2, p_4, p_5. Otherwise the robot has reached a critical placement of type 2. In both cases we claim that the robot has crossed a new trapezoid of $\mathcal{A}$: when the robot has got two legs on the end points of an edge p_4p_5 of P, the stability polygon must have changed which means that we have crossed an edge of $\mathcal{A}$, since we know the motion go ahead.

Suppose now that the robot placement is such that one of its legs is at maximal extension: G is on the circle $C(p_1)$. Let P' be the polygon obtained from P by deleting the critical site p_1.

Let the actual placement be p_1, p_2, p_3, with $G \in C(p_1)$. We check in $O(\log n)$ time if there exists a vertex p_4 of P' such that G is contained in the triangle

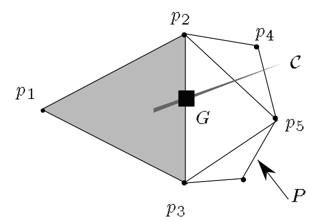

Figure 10: Exit from a stability-limit placement

p_2, p_3, p_4 (i.e. there exists a vertex of P' in the cone defined by G and the two lines s_2 and s_3: Figure 11.a). If so, we put the free leg on p_4 and lift the back leg on p_1. The motion then can go ahead until a new critical placement occurs.

Otherwise, call p_4, p_5 the vertices of P' lying on the line issued from p_1 and tangent to P' and let L_1 be the polygonal chain from p_4 to p_5 whose edges are edges of P' but not of P (Figure 9.b); on the chain L_1 we can perform a double binary search to find in $O(\log n)$ time two consecutive vertices p and q such that the triangle p_3, p, q contains G. It is clear, that p or q is on the same side of the line passing through p_3 and p_1 as p_2. Assume that this point is p. Then, since $G \in p_3pq$ but $G \notin p_3p_2q$, it is plain to observe that G must belong to triangle pp_1p_3.

The robot moves as follows: first the free leg is moved to p, the leg in p_2 is put on q, and then the back leg (resting in p_1) is lifted. The motion can then go on until a new critical placement occurs.

We have shown how the robot can pass a critical placement along the path $\mathcal{C}$: the complexity of the total traversal is given by the next theorems

Theorem 15 *A robot motion joining two given placements can be computed in $O(k|\mathcal{A}|\log n) = O(k^2 n \log n)$ time, if k is the density of the footholds. The motion consists of $O(k|\mathcal{A}| = k^2 n)$ leg moves.*

Theorem 16 *A robot motion along a shortest path joining two given placements can be computed in $O(|\mathcal{F}|^2 + |\mathcal{A}|\log n)$ time. The motion consists of $O(|\mathcal{A}|)$ leg moves.*

Proofs. For each critical placement we can find the new placement along $\mathcal{C}$ in time $O(\log n)$ since each search on the polygons can be done by binary search; in addition we have to compute dynamically the new convex hull at each intersection between $\mathcal{C}$ and an arc of $\mathcal{A}$, which can be done in $O(\log n)$ time for each intersection using an offline dynamic algorithm.[HS91] Since, by Proposition 12 (resp. 14) $\mathcal{C}$ crosses a given trapezoid of $\mathcal{A}$ only k (resp. a bounded number of) times, the convex hull maintenance is invoked at most $O(k|\mathcal{A}|)$ (resp. $|\mathcal{A}|$) times. Lastly we have to know how many critical placements

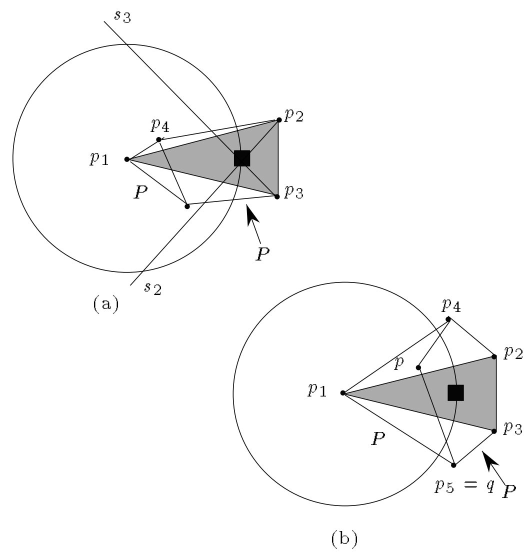

Figure 11: Exit from a maximal extension placement

occur along $\mathcal{C}$: leg maximum extension placements occur at most each time the trajectory crosses an arc, that is $O(k|\mathcal{A}|)$ (resp. $|\mathcal{A}|$). If we are in a stability limit placement, we have shown above that after at most two new placements we have to cross an arc of $\mathcal{A}$: thus the same bound applies to this case.

Our placement strategy gives an upper bound on the number of leg moves: we have seen that for crossing each trapezoid we need at most two leg changes. Thus in total the robot performs at most $O(k|\mathcal{A}|)$ (resp. $|\mathcal{A}|$) leg moves. $\square$.

6 Concluding Remarks

Moreover, $|\mathcal{F}|$ is usually much smaller than $|\mathcal{A}|$, even when $\mathcal{A}$ has quadratic size. It would be interesting to have an algorithm that construct $\mathcal{F}$ in time proportional to the actual size of $\mathcal{F}$, preferably to the size of $\mathcal{A}$.

A natural extension of the present work is to study other legged robots, for instance robots whose legs are attached at different points of the (no longer ponctual) body. For a given orientation of the robot, we can extend some of our results. We associate to each leg a color and color the footholds that a given leg can reach with the color of the leg (a foothold may have several colors). The definition of the stability region (Equation 3) must be modified : the role of the convex hull $CH(S)$ is now played by the union of the triangles whose vertices have distinct colors. It can be shown that such a union has linear size and can be computed in optimal $\Theta(n \log n)$ time.[BDP91] It follows that $\mathcal{F}$ has size $O(n|\mathcal{A}|)$ and can be computed in $O(n|\mathcal{A}| \log n)$ time. We do not know if these bounds are tight.

Further extensions under consideration will allow these robots to rotate, thus adding a third degree of freedom.

It would also be interesting to consider a more general setting involving 3-D space, obstacles and more complex footholds.

Acknowledgements

The authors would like to thank Jean-Pierre Merlet for supplying him with his marvellous interactive drawing preparation system JPdraw .

References

[AGSS87] A. Aggarwal, L. J. Guibas, J. Saxe, and P. Shor. A linear time algorithm for computing the Voronoi diagram of a convex polygon. In *Proc. 19th Annu. ACM Sympos. Theory Comput.*, pages 39–45, 1987.

[BDP91] J.-D. Boissonnat, O. Devillers, and F. Preparata. Computing the union of 3-colored triangles. *Internat. J. Comput. Geom. Appl.*, 1(2):187–196, 1991.

[BHK+89] J. Bares, M. Hebert, T. Kanade, E. Krotkov, T. Mitchell, R. Simmons, and W. Whittaker. Ambler, an autonomous rover for planetary exploration. *Computer*, pages 18–26, June 1989.

[HK91] S. Hirose and O. Kunieda. Generalized standard foot trajectory for a quadruped walking vehicle. *The International Journal of Robotics Research*, 10(1), February 1991.

[HNKU84] S. Hirose, M. Nose, H. Kikuchi, and Y. Umetani. Adaptive gait control of a quadruped walking vehicule. In *Int. Symp. on Robotics Research*, pages 253–277. MIT Press, 1984.

[HS91] J. Hershberger and S. Suri. Offline maintenance of planar configurations. In *Proc. 2nd ACM-SIAM Sympos. Discrete Algorithms*, pages 32–41, 1991.

[Lee82] D. T. Lee. On k-nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, C-31:478–487, 1982.

[PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, NY, 1985.

[RR84] Special issue on legged locomotion. *International Journal of Robotics Research*, 3(2), 1984.

[RR90] Special issue on legged locomotion. *International Journal of Robotics Research*, 9(2), April 1990.

[Sha91] M. Sharir. On k-sets in arrangements of curves and surfaces. *Discrete Comput. Geom.*, 6:593–613, 1991.

[Wel85] E. Welzl. Constructing the visibility graph for n line segments in $O(n^2)$ time. *Inform. Process. Lett.*, 20:167–171, 1985.