A Reachability-based planner for sequences of acyclic contacts in cluttered environments

S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, J. Pettré

Abstract Multiped locomotion in cluttered environments is addressed as the problem of planning acyclic sequences of contacts, that characterize the motion. In order to overcome the inherent combinatorial difficulty of this problem, we separate it in two subproblems: first, planning a guide trajectory for the root of the robot and then, generating relevant contacts along this trajectory. This paper proposes theoretical contributions to these two subproblems. We propose a theoretical characterization of the guide trajectory, named "true feasibility", which guarantees that a guide can be mapped into the contact manifold of the robot. As opposed to previous approaches, this property makes it possible to assert the relevance of a guide trajectory without explicitly computing contact configurations. Indeed, this property is efficiently checked using a low dimensional sampling-based planner (e.g. we implemented a visibility PRM). Since the guide trajectories that we characterize are easily mapped into a valid sequence of contacts, we then focus on how to select a particular sequence with desirable properties, such as robustness, efficiency and naturalness, only considered in cyclic locomotion so far. Based on these novel theoretical developments, we implement a complete acyclic contact planner and demonstrate its efficiency by producing a large variety of motions with three very different robots (humanoid, insectoid, dexterous hand) in five challenging scenarios. The quality of the obtained motions and the performance of the algorithm make it the first acyclic contact planner suitable for interactive applications.

Steve Tonneau, Nicolas Mansard
LAAS-CNRS, Toulouse, e-mail: `surname.lastname@laas.fr`

Chonhyon Park, Dinesh Manocha
UNC, Chapel Hill e-mail: `dm/chpark@cs.unc.edu`

Franck Multon
Unversité Rennes II e-mail: `fmulton@irisa.fr`

Julien Pettré
Inria e-mail: `julien.pettre@inria.fr`

1 Introduction

1.1 State of the art

We consider the problem of planning the acyclic sequence of contacts describing the motion of a multiped robot in a cluttered environment. Acyclic contact planning is a particular class of motion planning where every configuration of the resulting trajectory must be in contact with the environment in order to support the balance of the system. The difficulty of the problem comes both in practice from the proximity to the obstacles (that tends to make the sampling of valid configuration tedious) and in theory from the foliation of the configuration space, where zero-measure manifolds intersect in a combinatorial manner [18]. Acyclic motion planning is a problem of interest in robotics, neurosciences and biomechanics. It is also interesting for virtual character animation. Early contributions in this field rely on local adaptation of motion graphs [12] or ad-hoc construction of locomotion controllers [16]. These approaches can intrinsically not adapt to new situations or discover complex behaviors in unforeseen contexts. In robotics, the attention of the community first focused on the generation of cyclic locomotion patterns, in particular for bipedal walking on flat terrains [11]. While planning cyclic bipedal contacts is now mature, with existing real-time solutions [4], the problem remains open for more generic acyclic contacts.

The issue of planning acyclic contacts was first completely described in [7], where it is proven to require the handling of two simultaneous problems: $\mathscr{P}_1$: a relevant guide trajectory for the root of the robot in $SE(3)$; and $\mathscr{P}_2$: the planning of a discrete sequence of acyclic, balanced contact configurations along the trajectory[1]. A key issue is to avoid combinatorial explosion when considering at the same time the possible contact configurations and the potential trajectories. This seminal paper proposes a first effective algorithm, able to handle simple situations (such as climbing scenarios), but not scalable to arbitrary environments. Following it, several papers have applied this approach in particular situations, typically limiting the combinatorial by imposing a fixed set of possible contacts ([10], [19]).

After [7], most of the papers have explored alternative formulations to handle the combinatorial issue. Two main directions have been explored. **On one hand, local optimization of both the root trajectory $\mathscr{P}_1$ and the contact positions $\mathscr{P}_2$** has been used, to trade the combinatorial of the complete problem for a differential complexity, at the cost of local convergence. A complete example of the potential offered by such approaches is given in [14], with a successful adaptation to a real robot in [13]. To keep reasonable computation times, the method uses a simplified dynamic model for the avatar. Still, the computation time is far from interactive (about 1 minute of computation for a sequence of 20 contacts). In [8] contact planning is solved globally as a mixed integer problem, but only cyclic, bipedal locomotion is considered. Aside from the computation cost, a major drawback of these

[1] A third non trivial problem, $\mathscr{P}_3$, not adressed in this work, then consists in interpolating a complete motion between two postures of the contact sequence.

optimization based approaches is thus that they only offer local convergence when applied to acyclic contact planning.

On the other hand, the two problems $\mathscr{P}_1$and $\mathscr{P}_2$might be decoupled to reduce the complexity. The feasibility and interest of the decoupling is shown in [9], by manually setting up a rough guide trajectory (i.e. an ad-hoc solution to $\mathscr{P}_1$). $\mathscr{P}_2$is then addressed as the combinatorial computation of a feasible contact sequence in the neighborhood of the guide. A solution can then be found, at the cost of prohibitive computation times (several hours). Furthermore, this approach is suboptimal because the quality of the motion is conditioned by the relevance of the guide trajectory, which is not evaluated *a priori*. In [5], the authors precisely focus on automatically computing a guide trajectory with guarantees of contact feasibility, by extending key frames of the trajectory into whole-body, balanced contact configurations. Randomly sampled configurations are projected into the contact submanifold using a generalized inverse kinematics solver, a computationally expensive process (about 15 minutes are required to compute a guide trajectory in the examples presented). Moreover this explicit projection is yet an insufficient condition and does not provide strong guarantees on the feasibility of the path between two key positions in the trajectory.

1.2 Paper contribution and organization

Although the optimization approach is promising, we choose to focus on the sample-based methodology, more able to find complex trajectories in cluttered environments. While the theoretical structure of the problem is well understood, there is currently no scalable method to solve it. The combinatorial of the original problem (as described in [7]) is too high to have any hope of tackling it directly. Alternative formulations are necessary to obtain practical solutions. We believe that the separation, proposed in [9], between the guide trajectory and the contact sequence is the most promising direction. However, this direction raises two theoretical questions that remain to be solved, or even to be properly formulated:

- The guide trajectory must satisfy a property guaranteeing the existence of a contact sequence to actuate it[2]. This property has not been studied yet: the only way to validate a trajectory is to explicitly compute the contacts, which is computationally not reasonable, as shown in [5].
- There is an infinite combination of possible contact sequences for a given root trajectory. The selection of one particular contact sequence with interesting properties (minimum number of contact change, robustness, efficiency or naturalness) has been studied for cyclic cases, as in [10], but has not been efficiently applied to cluttered environments (e.g. [6] and [9] mostly randomly pick one contact sequence, leading to possibly very tedious contact sequences).

[2] This property is related to the controllability of the root actuated by the contact forces, but for discrete bounded properties.

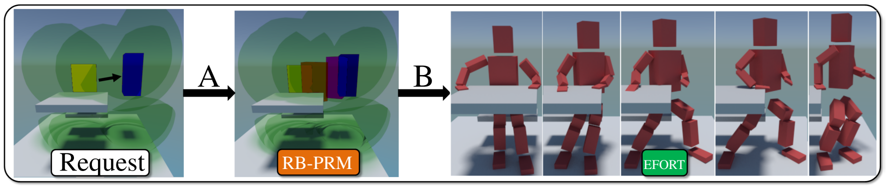

Fig. 1 Overview of our 2-stage framework. (A) Given a path request between the yellow and blue positions, a guide trajectory is computed in C_{reach} using RB-PRM. (B) The trajectory is extended into a discrete sequence of contact configurations using EFORT.

We claim that the desirable contact properties of a guide trajectory, proposed in [5], can be formulated in a space of lower dimension, which we call C_{reach}. This formulation can make the planning of a guide trajectory more efficient computationally, while providing equivalent guarantees to planning directly in the configuration space. Among the particular properties obtained when planning in C_{reach}, we would like to guarantee that any reduced trajectory can actually lead to a feasible sequence of contacts, in which case we say that the reduced trajectory is truly feasible. It is possible in theory to guarantee that any reduced trajectory is truly feasible, even if it is more efficient in practice to approximate this property. The true-feasibility of the guide trajectory then allows us to focus on the selection of one particular sequence of contacts, for example one that minimizes the number of contacts in the sequence or maximizes the robot efficiency or style.

Based on these fundamental observations, we implement a very efficient acyclic contact planner. Our method is based on a probabilistic roadmap (PRM), that computes offline guide trajectories that are approximately truly feasible. The planner then resolves online the contact sequence by refining a guide trajectory computed from the PRM. Our planner is able to compute physically-consistent contact sequences for very complex systems (a humanoid, 28 joints; and an insectoid, 48 joints) in a few seconds for classical scenarios like climbing, and less than a minute for very complex problems like egress from a damaged truck. The planner also generalizes to planning dexterous manipulation movements, as demonstrated by preliminary results.

The contributions of the paper are twofold. We propose the first theoretical characterization of today's most efficient practical approach to sampled-based planning of acyclic contacts. And based on this characterization, we propose a very efficient and general implementation of an acyclic contact planner, the first one compatible with interactive applications.

We propose a framework to address the motion planning problem for multiped robots in cluttered environments: given a start and a goal configuration, the objective is to compute a sequence of contact configurations allowing to achieve the motion. For instance, we can consider the task of standing up, illustrated in Figure 1–right. The problem is decoupled into two sequential phases: 1) the computation of a guide trajectory for the root of the robot; 2) the computation of a discrete sequence of contact configurations allowing to achieve the motion along the trajectory. The re-

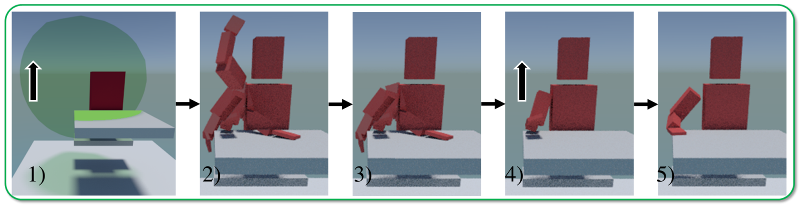

Fig. 2 Generation of a contact configuration for the right arm of a humanoid robot. 1) Selection of reachable obstacles. 2) A request is performed on a database of configurations. 3) Configurations too far from contact are eliminated. 4) The best candidate according to EFORT is chosen. 5) The final contact is achieved using inverse kinematics.

mainder of this section presents the general organization of our method in Section 2. The two following sections 2 and 3 present respectively our answer to problems $\mathscr{P}_1$and $\mathscr{P}_2$. Finally, we propose a complete experimental validation of the planner with three very different kinematic chains (humanoid, insectoid and three-finger manipulator) in various scenarios.

1.3 Computation of a guide trajectory

We first consider the problem of planning a relevant guide trajectory. The objective is to compute a trajectory of root placements which will allow contact creation. The objective of this first part is to preserve the completeness of the planner: it should be able to explore any possible guide trajectory, but at the same time, any computed guide trajectory must be truly feasible, i.e. must lead to a valid sequence of contacts.

An intuitive description of such placements is ”close, but not too close”: close, because a contact surface must be partially included in the range of motion of the robot (represented for the right arm in Fig. 2–1); not too close, because the robot must avoid collision (which is represented by the hull including the torso in Fig. 2–1). We define formally C_{reach}, the set of interesting root placements, in which we compute a guide trajectory with a sampling based planner, the reachability PRM – RB-PRM– (Figure 1–A). Planning in C_{reach} boils down to planning in SE(3), which has an acceptable practical complexity. Details are presented in Section 2.

1.4 Generating a discrete sequence of contact configurations

The second stage is to extend the guide trajectory into a sequence of contact configurations (Fig. 1–B). Thanks to the nice property that we manage to obtain for the guide, obtaining a random sequence of contact is an easy problem. The goal of this part is to select an efficient sequence, in particular by reducing the number of contacts in the sequence. To create contacts in an efficient manner, we consider

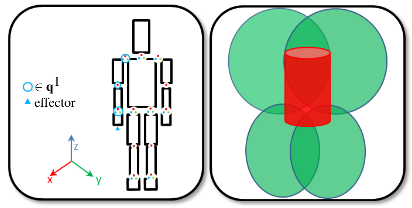

Fig. 3 Left: Robot in a rest configuration. The right arm is denoted as the limb R^1. Each colored dot represents a degree of freedom around an axis. Right: Volumes of the robot. The red geometry denotes W^0 and must remain collision-free. The green spheres are the W^k.

each limb as a manipulator attached to the root, and select the most relevant contact from a database of precomputed configurations (Figure 2). The relevance is defined as the contribution to the quasi static balance of the robot, and as the contribution to the motion of the root. This task efficiency is measured based on the Extended FORce Transmission ratio (EFORT), that we proposed in [20]. Details are presented in Section 3.

1.5 Notation conventions and definitions

A vector $\mathbf{x}$ is denoted with a bold lower case letter. A matrix $\mathbf{A}$ is denoted with a bold upper case letter. A set C is denoted with a upper case italic letter. Scalar variables and functions are denoted with lower case italic letters, such as r or $f(\mathbf{x})$.

A robot is a kinematic chain R, comprising $n+6$ degrees of freedom (DOFs). R is composed of l limbs $R^k, 1 \leq k \leq l$, attached to a root. It is described by a configuration $\mathbf{q} \in SE(3) \times \mathbb{R}^n$. We define some relevant projections of $\mathbf{q}$:

- $\mathbf{q}^k$ denotes the configuration (a vector of joint values) of the limb R^k (Fig. 3);
- $\mathbf{q}^{\bar{k}}$ denotes the vector of joint values of R **not** related to R^k. We define for convenience $\mathbf{q} = \mathbf{q}^k \oplus \mathbf{q}^{\bar{k}}$;
- $\mathbf{q}^0 \in SE(3)$ denotes the position and orientation of the root of the robot R.

The environment O is defined as the union of the obstacles O_i it contains. The volume encompassing the trunk of the robot is denoted W^0 (Fig. 3-right: central cylinder). The range of motion of a limb R^k is denoted W^k (Fig. 3-right: the four ellipses).

$$W^k = \left\{ \mathbf{x} \in \mathbb{R}^3 : \exists \mathbf{q}^k, \mathbf{p}^k(\mathbf{q}^k) = \mathbf{x} \right\} \tag{1}$$

where $\mathbf{p}^k$ denotes the end-effector position of R^k (translation only) for $\mathbf{q}^0$ the null displacement. We also define $W = \bigcup_{k=1}^{l} W^k$. Finally, we define $W^k(\mathbf{q}^0), 0 < k \leq l$ as the volume W^k translated and rotated by the rigid displacement $\mathbf{q}^0$.

2 Computation of a guide trajectory in C_{reach} (Stage 1)

We consider the problem of computing a guide trajectory $\mathbf{q}^0(t) : [0,1] \longrightarrow SE(3)$ for the geometrical root of a multiped robot, connecting start and goal configurations. As said in previous section, the goal is to enforce that any configuration $\mathbf{q}^0$ of the guide is truly feasible, i.e. can be mapped to a balanced configuration in contact. We denote by $C_{contact} \subset SE(3) \times \mathbb{R}^n$ the contact submanifold of the robot.

We say that a root placement $\mathbf{q}^0$ is *truly feasible* if there exists a bijective mapping[3] π such that

$$\pi : \quad \mathbf{q}^0 \in SE(3) \longrightarrow \mathbf{q}^0 \oplus \mathbf{q}^{\overline{0}} \in C_{contact} \tag{2}$$

The set of all truly feasible root placements is denoted by C_{reach}. By extension, a trajectory $\mathbf{q}^0(t)$ is truly feasible if $\forall t \in [0,1], \mathbf{q}^0(t) \in C_{reach}$.

For a two-stage acyclic contact planner to be exact and complete, we need the combination of two conditions on a guide trajectory generator: all the generated trajectories must be *truly feasible* (sufficient condition); the guide planner must be complete, i.e. it must be able to discover any truly feasible trajectory (necessary condition)[4].

2.1 Conditions for true feasibility

By default, the true feasibility implies a constructive demonstration by exhibiting a valid π. This is the approach chosen by [5]. However, computing a valid $\mathbf{q}^{\overline{0}}$ is costly in practice. In this section we rather define a necessary condition and a sufficient condition for true feasibility that do not require this explicit computation.

True feasibility: necessary condition

For a contact to be possible, a volume $O_i \in O$ necessarily intersects with the range of motion $W(\mathbf{q}^0)$ (Fig. 2–1). Furthermore, if $\mathbf{q}^0$ is truly feasible, then the trunk of the robot $W^0(\mathbf{q}^0)$ is necessarily not colliding with the environment O.

Therefore we can approximate C_{reach} with a set $C_{reach} \subset C^1_{reach}$ with the reachability condition defined as:

$$C^1_{reach} = \{\mathbf{q}^0 : W(\mathbf{q}^0) \cap O \neq \emptyset \wedge W^0(\mathbf{q}^0) \cap O = \emptyset\} \tag{3}$$

It is straighforward to prove that $C_{reach} \subset C^1_{reach}$ (by construction of the included set). This inclusion is very important: it directly implies that any motion-planning

[3] This mapping is not uniquely defined.

[4] The proof is immediate, using a contradiction approach.

algorithm with a guaranty of completeness in $SE(3) \times \mathbb{R}^n$ is complete in $C_{reach} \times \mathbb{R}^n$. This is a strong reduction of the search space, which can be directly applied to any existing method.

The condition C^1_{reach} is only necessary which means that one such root placement might not be truly feasible: in practice it is not guaranteed to find a valid sequence of contacts for every guide trajectory in C^1_{reach}.

Sufficient condition for true feasibility

A trivial sufficient condition for true feasibility can be constructed as a variation of C^1_{reach}, by replacing W^0 with a bounding volume B^{max} encompassing the whole robot in a given pose, except for the effector surfaces to be in contact[5]. We denote by $C^{\infty}_{reach} \subset C_{reach}$ the set of root placements corresponding to the sufficient condition.

In general, the inclusion is strict, which means that we lose the completeness of the two-stage contact planner (i.e. the planner is not able to discover a trajectory inside $C_{reach} \setminus C^{\infty}_{reach}$). However, the sufficient condition guarantees that any such trajectories leads to a valid sequence of contacts (i.e. π is defined).

2.2 Reachability: a compromise condition

The sufficient condition is not interesting in practice since it leads the solver to lose too many interesting trajectories. The necessary condition is not perfect either, since the first stage of the planner would stop on a guide that is not truly feasible in practice. It might be possible to find a shape B that is necessary and sufficient; however, it seems intuitively very unlikely in general. The construction of a shape $W^0 \subset B \subset B^{max}$ leading to a necessary and sufficient condition (or the proof of its inexistence) is out of the scope of this work.

However between W^0 and B^{max}, a trade-off can be found between a necessary and a sufficient condition. We define W^0_s as the volume W^0 subject to a scaling transformation by a factor $s \in \mathbb{R}^+$. We then consider the spaces C^s_{reach}

$$C^s_{reach} = \{\mathbf{q}^0 : W(\mathbf{q}^0) \cap O \neq \emptyset \wedge W^0_s(\mathbf{q}^0) \cap O = \emptyset\} \tag{4}$$

The higher s is, the closer the reachability condition is to being sufficient, and if $s = 1$, the planner is complete. The parametrization of s allows to find a trade-off between these two desirable properties. Section 4 shows that in practice, it is easy to adjust s to keep most of the interesting guides without introducing incorrect guides.

---

[5] This condition is trivial in the sense that the resulting W has a zero measure. For the need of the proof, the trivial sufficient condition is enough. In practice, the construction of a non-trivial including shape W^0 was possible for all the robot structures we considered.

2.3 Computing the guide trajectory in C^s_{reach} with RB-PRM.

Once a value of s has been fixed, any sampling-based motion planner can be used to plan a trajectory in C^s_{reach}. The only variation consists in replacing the classical collision checking method with a test of appartenance to C^s_{reach} when verifying that configurations and associated local paths are valid. For this reason, there is no need to provide a pseudo-code for RB-PRM, although we provide here relevant information about our own implementation.

We have chosen to implement RB-PRM as a variation of Visibility-PRM [15], which usually leads to a smaller set of nodes than classical PRM planners. The associated drawback is that the paths returned by the planner might not be the shortest ones, which is typically not an issue in highly cluttered environments.

To sample more efficiently configurations of C^s_{reach}, we bias the sampling process to generate near obstacles configurations, similarly to [1] to generate configurations in narrow passages. First, a configuration is set to a random point on the surface of one obstacle. The configurations are then translated and rotated randomly until the reachability condition is satisfied. Again, our implementation only differs in the fact that C^s_{reach} is sampled instead of C_{free}.

3 From a guide trajectory to a discrete sequence of contact configurations (Stage 2)

As an input of this stage, we consider a truly feasible guide trajectory $\mathbf{q}^0(t)$: $[0,1] \longrightarrow SE(3)$ for the root of the robot R. We now consider the second problem of computing a trajectory $\mathbf{q}^{\overline{0}}(t)$ for the limbs of the robot. Since we assume true feasibility, we know that such a trajectory exists. Contrary to previous works [9, 5], the goal here is not to find any such trajectory but rather to select one with particular properties. Specifically, we show here how to build a contact sequence with a small number of contact variations and good-efficiency and naturalness of the postures.

More precisely, any mapping π introduced in (2) can be used to expand $\mathbf{q}^0(t)$ into a whole-body trajectory. We propose here a particular construction of π leading to interesting contact sequences.

3.1 Extension of the guide trajectory

The guide trajectory $\mathbf{q}^0(t)$ is first discretized into a sequence of j key placements:

$$\mathbf{Q}^0 = [\mathbf{q}^0_0; \mathbf{q}^0_i; ..., \mathbf{q}^0_{j-1}]$$

where $\mathbf{q}^0_0$ and $\mathbf{q}^0_{j-1}$ respectively correspond to the start and goal configurations. To ensure continuity in the contact transition phases, we rewrite π under the following

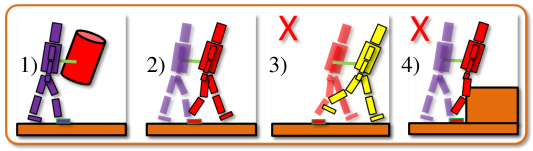

Fig. 4 Contacts are maintained unless their position is too far, or the environment prevents it.

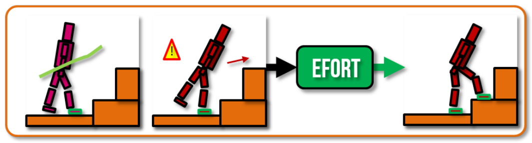

Fig. 5 Contacts are generated when the configuration is not balanced.

recursive form for any $0 < i < j$:

$$\pi : \begin{cases} \mathbf{Q}^0 \in C_{reach} \longrightarrow C_{contact} \\ \qquad\qquad \mathbf{q}_i^0 \longrightarrow \mathbf{q}_i^0 \oplus g(\mathbf{q}_{i-1}, \mathbf{q}_i^0) \end{cases}$$

We initialize the recurrence with $\pi(\mathbf{q}_0^0) = \mathbf{q}_0$ the initial configuration of the robot. The function g is defined independently by g^k for each limb R^k. In defining g^k, two aspects must be considered. Is the limb R^k in contact? And which criteria is it optimizing?

Maintaining a contact: If possible, a limb in contact at time $i-1$ remains in contact at i. The contact is broken if an inverse kinematics solver fails to find a collision free limb configuration which satisfies joint limits. The solver is a direct implementation of the inverse kinematics solver with task priorities framework proposed in [3], which handles explicitely joint limits, and allows to integrate collision avoidance as a secondary criterion.

If the solver fails, the contact is broken and a collision free configuration is assigned to the limb.

Once a first candidate configuration is taken for all limbs, the quasi-static balance is tested by whether the weight wrench is in the gravito-inertial cone (i.e. there exists valid contact forces that compensate for the weight of the robot), using the geometric approach described in [17]. If the balance is not obtained, new contacts are randomly generated using the following procedure.

Creating a contact: We consider a configuration where some limbs are in contact, some are free and quasi-static balance is not enforced. To enforce balance, we proceed in the following manner: we randomly select a contact free limb; if there is no contact free limb, we select the limb that made contact first. Using the contact generator introduced in [20], we project the configuration of this limb into a contact that enhances balance, if it exists (Figure 5); If balance is not achievable and a contact is possible, it is generated anyway; If balance is not achieved, the next limb is selected and projected into a contact configuration, and so on. This approach can lead to the repositioning of existing contacts, in which case intermediate states are inserted to reposition the contacts. This current implementation does not guarantee that the planner will succeed in generating a balanced configuration, because true feasibility is not fully guaranteed. However in practice the planner is successful in the large majority of cases, as discussed in section 4.2.

3.2 Contribution to the global movement: the EFORT criteria

EFORT criterion: If only relying on the random sampling to select new contacts, the planner produces inefficient postures. The resulting contact sequence is then poorly efficient and unnatural. Moreover, the limbs are not well configured and are not able to efficiently follow the general movement: contacts break frequently.

When creating additional contacts, we therefore propose to select particular configurations that allow to exert a force compatible with the direction of motion. This task efficiency is measured with the Extended FORce Transmission ratio (EFORT) [20]. The measure of EFORT is given by

$$\alpha_{EFORT}(\mathbf{q}^k, \mathbf{m}) = [\mathbf{m}^T(\mathbf{J}\mathbf{J}^T)\mathbf{m}]^{-\frac{1}{2}}(\nu_0 \mathbf{n}^T \mathbf{m}) \tag{5}$$

where $\mathbf{J}$ is the Jacobian matrix of the limb R^k in configuration $\mathbf{q}^k$; ν_0 is the friction coefficient of the contact surface; $\mathbf{n}$ is the normal of the contact surface; and $\mathbf{m}$ is the direction opposite to the motion, given by the 3D vector connecting $\mathbf{q}_i^0$ and $\mathbf{q}_{i+1}^0$. The first part of the equation measures the ratio between the joint torques and the resulting force applied along $\mathbf{m}$. The second part quantifies the odds of slipping while applying a force along $\mathbf{m}$.

Optimization at creation: In practice, a database of configurations is stored for each limb, which can be considered as manipulator arms. The database is implemented as an octree data structure, indexed by the end-effector positions of the configurations (and additionally storing $\mathbf{J}$). Upon request, the octree returns a set of configurations close to contact (Fig. 2-3). These candidates are sorted based on their task efficiency, given by α_{EFORT}. The first candidate in this list satisfying the balance criterion and is collision free is selected and projected on the contact surface using our inverse kinematics solver.

4 Results

The main strength of our planner is that it efficiently works for arbitrary robot shapes. We first validate this aspect by producing a large variety of movements with three very different robots (humanoid, insectoid, dexterous hand) in five challenging scenarios. Two evaluations of the method are provided: qualitatively, by displaying

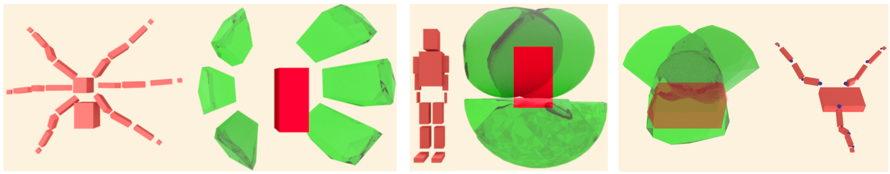

Fig. 6 Robots and associated volumes: in red W_s^0; in green the range of motion of each limb.

the naturalness of the contact sequence in the companion video; and quantitatively by statistically measuring the validity of the compromise condition (Sec. 4.2) and the performances of the algorithm (Sec. 4.3).

4.1 Robot models and scenarios

Fig. 6 describes the robot used in the experiments. The humanoid robot has four limbs, each with 7 DOFs. It has a total of 34 DOFs. The insectoid robot has six limbs, each with 8 DOFs, and a total of 54 DOFs. The hand has three fingers, each with 6 DOFs and a total of 24 DOFs.

In all the scenarios considered, the formulation of the problem is always the same: a start and goal configuration are provided as an input of the scenario, and the framework outputs a sequence of statically balanced contact configurations connecting the start and goal configurations. A companion video available at `http://youtu.be/LmLAHgGQJGA` (anonymous link) displays the complete contact sequence obtained in all these scenarios. The video only renders the contact configurations (not the interpolation between contacts, which is out of the scope of the paper).

Truck egress (humanoid and insectoid): The robot must leave a truck the doors of which are blocked: it has to crawl through the front window. Figure 7 presents the sequence of contacts obtained for both robots: RB-PRM can find solutions in highly cluttered environments with narrow passages.

Climbing (humanoid and insectoid): The robot has to climb on a wall with several grasps disposed along it. In this scenario, we give stronger conditions for the sampled root placements: we require that more than one range of motion W^k collide with obstacles of the environment. Fig. 8 presents the contact sequence obtained for the humanoid robot.

Manipulation of a pen (3-finger hand): This scenario is proposed to illustrate the genericity of our approach: we consider a manipulation task for a robotic hand and use our contact planner to compute a guide trajectory for the fingers, considered as effectors (Figure 9). Although we do not address the hard issue of accounting for rolling motions, the planner is able to compute the shown sequences, this in less than 5 seconds.

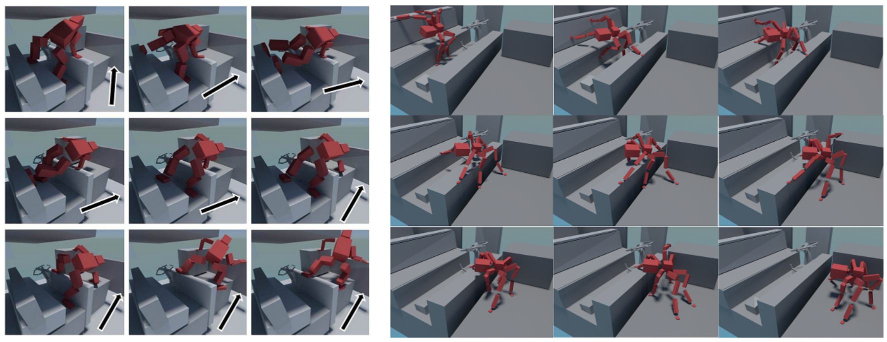

Fig. 7 The computed contact sequences for the truck egress scenario. Only selected postures are shown for the insect.

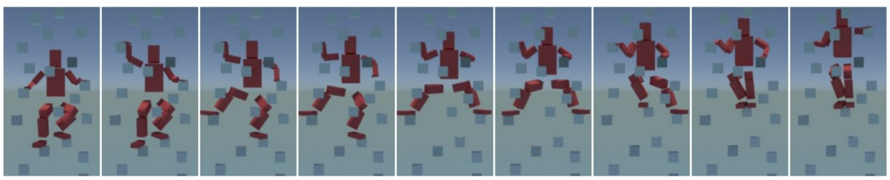

Fig. 8 The computed sequence for the climbing scenario.

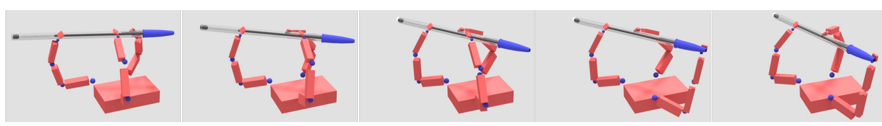

Fig. 9 Contact sequence found for a pen manipulation in a zero gravity environment.

Other scenarios (humanoids): The standing-up scenario (already presented in Fig. 1) is a setup taken from [9]: it corresponds to a long narrow passage in the configuration space. In the crouching scenario, demonstrated in the companion video, the character automatically goes from a standing to a crouching position to crawl under an obstacle.

4.2 Parametrization of the reachability condition

To find the appropriate C^s_{reach} in which to sample the guide trajectory, we computed the rejection rate for various values of s for each robot in the most cluttered truck scenario. For a given value of s, 10^6 root positions and orientations are computed in C^s_{reach}. In each case we try to generate a collision free contact configuration, with a

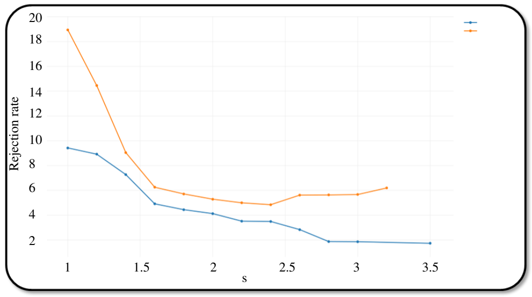

Fig. 10 Truck scenario rejection rates(%) for the humanoid(orange) and insectoid(blue), given s.

database comprising $N = 10^5$ sample configurations for each limb. The rejection rate is the ratio between the number of failures and the number of trials. From Figure 10 s is empirically chosen as the smallest value for which the rejection rate is minimal. For the humanoid, we thus chose $s = 2.2$, and for the insect, $s = 2.8$.

4.3 Performance

The number of samples used for generating the contacts of each limb is 10000. Table 1 presents the average time (seconds) spent in the phases of the planner, for each phase and each scenario, and the number of contact phases of the sequence.

We observe that many contacts are required for the insect, which can be explained by its restricted range of motion. The time spent generating the navigation graph is about one minute. The time spent in generating the graph of the climbing scenario, despite the relatively open environment, is explained by the additional restrictions imposed on the reachability condition. The difficulty to find a balanced configuration essentially influences the time spent generating the contacts.

The number of contacts in the sequence gives a rough estimation of its duration in seconds. Except for the robots crawling out of the truck, all the contact generation

	Generate RB-PRM (offline)	Generating the contact sequence	Number of contact states
Truck egress (humanoid)	73	15	10
Truck egress (insectoid)	70	23	48
Climbing (humanoid)	25	5	15
Climbing (insectoid)	21	27	51
Crouching (humanoid)	5	6	22

Table 1 **Average time** (in seconds) spent in RB-PRM generation, and the online generation of the contact sequence.

are real-time. Additionaly to the quality of the generated trajectories shown in the video, these computation times are a major practical achievement.

5 Discussion and future work

In this paper we consider acyclic contact planning in cluttered environments, formulated as two sub problems that we address sequentially: $\mathscr{P}_1$: the computation of a guide trajectory for the root of the robot that can be extended ; $\mathscr{P}_2$: the computation of a discrete sequence of contacts along this trajectory. Our contribution to $\mathscr{P}_1$is a generic characterization of the properties that the guide trajectory must staisfy, in particular to enforce the completeness of the acyclic contact planner. We introduced a low dimensional space C_{reach} that can be mapped into the contact submanifold of the robot, approximated and efficiently sampled by our Reachability-Based planner. Our contribution to $\mathscr{P}_2$is a pragmatic contact generation scheme that can take into account criteria to enforce interesting properties on the generated contacts (such as robustness, energy efficiency or naturalness). One such criterion, EFORT, is used to demonstrate the method and optimizes a force exertion compatible with the direction of motion.

Aside from the theoretical contributions, our results demonstrate that our method allows a very interesting compromise between three criteria that are hard to conciliate: generality, performance, and quality of the solution, making it the first acyclic contact planner compatible with interactive applications. **Regarding generality**, the reachability condition, coupled with an approach based on limb decomposition, allows the method to address arbitrary multiped robots. The only pre-requisite is the specification of the volumes W^0 which can is adjusted from a statistical analysis such as the one run in section 4.2. **Regarding performance**, our framework outperforms existing methods in addressing either $\mathscr{P}_1$or $\mathscr{P}_2$, leading to computation costs close to real-time in statically known environments. **Regarding the quality of the trajectories**, a parametrization of the reachability condition allows us to compute relevant trajectories in all the scenarios presented, with low rejection rates. As for [5], failures can still occur, due to the compromise criterion used in computing the guide trajectory.

Future work will focus on a more accurate formulation of C_{reach} to address this limitation. Additionally, we will work on theoretically characterizing the benefits of using the EFORT criterion when generating the sequence (our understanding is yet only intuitive), and quantifying the reward in terms of energetic efficiency or naturalness. Generating the complete motions, by interpolating between the computed contact sequences, is of course our next objective. This is a mandatory step before going on the real robot. A reasonable approach is to search the optimal trajectory connecting two postures of the sequence, under the constraint of maintaining balance. In particular, we will consider optimization schemes based on reduced dynamic models, possibly extending the preliminary work of [2].

References

1. Amato, N.M., Burchan, O., Lucia, B., Dale, K., Jones, C., Vallejo, D.: Choosing good distance metrics and local planners for probabilistic roadmap methods. In: Proc. IEEE Int. Conf. Robot. Autom. (ICRA) (1998)
2. Audren, H., Vaillant, J., Kheddar, A., Escande, A., 1, K.K., Yoshida, E.: Model preview control in multi-contact motion–application to a humanoid robot. In: IEEE/RSJ Int. Conf. on Int. Robots and Syst. (IROS) (2014)
3. Baerlocher, P., Boulic, R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. The Visual Computer **20**(6) (2004). DOI 10.1007/s00371-004-0244-4
4. Baudouin, L., Perrin, N., Moulard, T., Lamiraux, F., Stasse, O., Yoshida, E.: Real-time replanning using 3D environment for humanoid robot. In: IEEE-RAS Int. Conf. on Humanoid Robots (Humanoid'11). Bled, Slovenia (2011)
5. Bouyarmane, K., Escande, a., Lamiraux, F., Kheddar, a.: Potential field guide for humanoid multicontacts acyclic motion planning. 2009 IEEE Int. Conf. on Robot. and Auto pp. 1165–1170 (2009). DOI 10.1109/ROBOT.2009.5152353
6. Bouyarmane, K., Kheddar, A.: Multi-Contact Stances Planning for Multiple Agents. In: ICRA'11: Int. Conf. on Robot. and Auto. Shanghai Int. Conf. Center, Shanghai, Chine (2011)
7. Bretl, T., Rock, S., Latombe, J.C., Kennedy, B., Aghazarian, H.: Free-climbing with a multi-use robot. In: M.H.A. Jr., O. Khatib (eds.) ISER, *Springer Tracts in Advanced Robot.*, vol. 21, pp. 449–458. Springer (2004)
8. Deits, R., Tedrake, R.: Footstep planning on uneven terrain with mixed-integer convex optimization. In: 14th IEEE-RAS Int. Conf. on Humanoid Robots, Humanoids 2014, Madrid, Spain, November 18-20, 2014 (2014)
9. Escande, A., Kheddar, A., Miossec, S., Garsault, S.: Planning Support Contact-Points for Acyclic Motions and Experiments on HRP-2. In: O. Khatib, V. Kumar, G.J. Pappas (eds.) ISER, *Springer Tracts in Advanced Robot.*, vol. 54, pp. 293–302. Springer (2008)
10. Hauser, K., Bretl, T., Harada, K., Latombe, J.C.: Using motion primitives in probabilistic sample-based planning for humanoid robots. In: S. Akella, N.M. Amato, W.H. Huang, B. Mishra (eds.) WAFR, *Springer Tracts in Advanced Robot.*, vol. 47, pp. 507–522. Springer (2006)
11. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point. In: IEEE Int. Conf. Robot. and Auto (ICRA). Taipei, Taiwan (2003)
12. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. In: ACM Trans. on Graphics, vol. 21. ACM, New York, NY, USA (2002)
13. Mordatch, I., Lowrey, K., Todorov, E.: Ensemble-CIO: Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids (2015)
14. Mordatch, I., Todorov, E., Popović, Z.: Discovery of complex behaviors through contact-invariant optimization. ACM Trans. on Graphics **31**(4) (2012)
15. Nissoux, C., Siméon, T., Laumond, J.: Visibility based probabilistic roadmaps. In: Proceedings 1999 IEEE/RSJ Int. Conf. on Int. Robots and Syst. Hum. and Environment Friendly Robots with High Int. and Emot. Quot., October 17-21,1999, Hyundai Hotel, Kyongju, Korea (1999)
16. Pettré, J., Laumond, J.P., Siméon, T.: A 2-stages locomotion planner for digital actors. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '03. Eurographics Association (2003)
17. Qiu, Z., Escande, A., Micaelli, A., Robert, T.: Human motions analysis and simulation based on a general criterion of stability. In: Int. Symposium on Digital Human Modeling (2011)
18. Siméon, T., Laumond, J., Cortes, J., Sahbani, A.: Manipulation planning with probabilistic roadmaps. Int. Journal of Robot. Research **23**(7-8) (2004)
19. Stilman, M.: Global Manipulation Planning in Robot Joint Space With Task Constraints. IEEE Trans. on Robot. **26**(3) (2010)
20. Tonneau, S., Pettré, J., Multon, F.: Using task efficient contact configurations to animate creatures in arbitrary environments. Computers & Graphics **45**(0) (2014)