Homotopic particle motion planning for humanoid robotics

Andreas Orthey, Vladimir Ivan, Maximilien Naveau, Yiming Yang, Olivier Stasse, Sethu Vijayakumar

Homotopic particle motion planning for humanoid robotics

Andreas Orthey[1], Vladimir Ivan[2], Maximilien Naveau[1], Yiming Yang[2], Olivier Stasse[1], Sethu Vijayakumar[2]

***Abstract*—Exploiting structure is essential to an understanding of motion planning. Here, we exploit the topology of the environment to discover connected components. Inside a connected component, instead of planning one trajectory in configuration space, motion planning can be seen as optimizing a set of homotopically equivalent particle trajectories. In this paper, we will concentrate on the problem of motion planning for a humanoid robot. Our contributions are: i) finding the homotopy classes of a single footstep trajectory in an environment, ii) finding a single footstep trajectory in a single homotopy class formulated as a convex optimization problem, and iii) finding a feasible upper body trajectory given a footstep trajectory, formulated as a set of convex optimization problems. This view provides us with important insights into the difficulty of motion planning, and – under some assumptions – allows us to provide the number of local minima of a given motion planning problem. We demonstrate our approach on a real humanoid platform with 36-dof in a highly restricted environment.**

I. Introduction

Motion is one of the fundamental building blocks of intelligent behavior, and a deep understanding would enable robots to achieve autonomous tasks in eldery care, deep space exploration, and nuclear waste removal.

We study here the motion planning problem $A = \{\mathcal{R}, \mathcal{C}, q_I, q_G, \mathbf{E}\}$ [1] with $\mathcal{R}$ be a robotic system, $\mathcal{C}$ the configuration space of $\mathcal{R}$, $q_I \in \mathcal{C}$ the initial configuration, $q_G \in \mathcal{C}$ the goal configuration and $\mathbf{E}$ the environment.

The motion planning problem was shown to be NP-hard [2], and for humanoid robots, computational time can become several hours in a narrow environment [3], [4], [5]. We argue that the main problem is the reliance on random sampling techniques [6]: if the subset of feasible configuration gets arbitrarily small, the convergence rate of random sampling gets arbitrarily high [7]. While random sampling is excellent for solving the problem in general, we argue here that to design truly efficient algorithms, we need to study, understand and exploit the underlying structure of the motion planning problem.

Here, we concentrate on investigating and exploiting the environment structure by extracting homotopy classes. A homotopy class is a set of functions, which can be continuously deformed into each other, as depicted in Fig. 1. For each homotopy class, we consider the trajectories of a set of particles $\{\tau_k\}_{k=0}^{\eta}$ on the robot body moving through $\mathbb{R}^3$ on space curves of the form $\tau_k : [0,1] \to \mathbb{R}^3$. We assume here that all particle trajectories are homotopically equivalent. Motion planning can then be conducted in the environment by first finding a single particle trajectory, and then finding all particles on the robot body by restricting them to belong to the same homotopy class as the single particle trajectory.

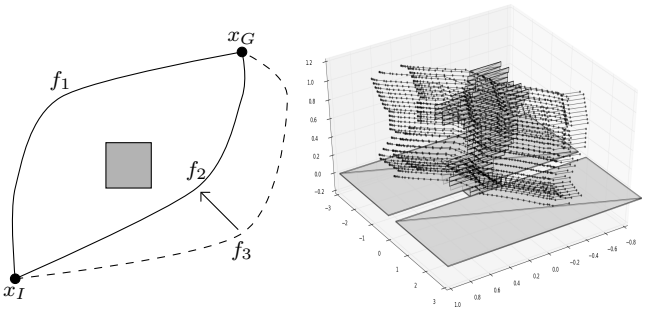


Fig. 1. **Left**: Two functions are in the same homotopy class, if they can be continuously deformed into each other, while fixing their end points. f_1 would be not homotopically equivalent to f_2, while f_3 would be. **Right**: In 3d, we conduct homotopic motion planning for a set of particles, particles with homotopically equivalent space curves.

Towards this goal, we decompose the open space of the environment into smaller volumes and analyze their covering to compute homotopy classes of robot particles moving through open space. We argue here that performing motion planning locally in one homotopy class ensures continuity, which is a requirement for optimization based planners. This decomposition of motion planning is called homotopic motion planning [8].

Our contributions are

- Identification of the homotopy classes in a given environment for a sliding footstep and the approximation of the free space
- Formulation of the problem of finding a sliding contact trajectory in one homotopy class as a convex optimization problem
- Formulation of the problem of finding a set of particle trajectories on the robot body, which are constrainted by the contact trajectory, as a set of convex optimization problems

Sec. II describes how we we decompose an environment into walkable surfaces, homotopy classes and the free space inside of one homotopy class. Sec. III formulates optimizing

* This research has received funding from the European Union Seventh Framework Programme (FP7/2007 - 2013) under grant agreement n° 611909, KoroiBot.

[1] are with the CNRS, LAAS,7 av. du Colonel Roche, F-31400, Toulouse, France, Univ de Toulouse, LAAS, F-31400, Toulouse, France {aorthey,mnaveau,ostasse} at laas.fr

[2] are with the University of Edinburgh, Edinburgh, UK {vivan,yiming.yang,sethu.vijayakumar} at ed.ac.uk

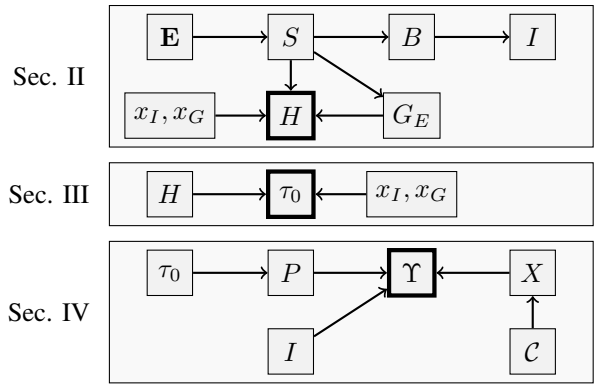


Fig. 2. The conceptual overview about this paper. **Top**: in section II we decompose the environment $\mathbf{E}$ into walkable surfaces S, intersections I and intersections I. From the start contact x_I, the goal contact x_G and the connectivity graph G_E we compute the homotopy classes H on S. **Middle**: Given the homotopy classes H, the start contact x_I and the goal contact x_G, we compute a sliding footstep trajectory τ_0, supported on H. **Bottom**: We compute upper-body particle trajectories Υ from a given footstep trajectory τ_0, planes P, cuboids B, and from cross-sections X generated by sampling robot configurations $\mathcal{C}$.

a single footstep trajectory as a convex optimization problem, and Section IV formulates the upper body optimization as a set of convex optimization problems under convex inequality constraints from the environment. The reader is refered to consult Fig. 2 for a technical overview.

II. Environment Homotopy Decomposition

In this section, we describe how we compute the free space of a given environment, and its connectivity. We start by reasoning about surfaces on which a foot contact is possible, which we call *walkable surfaces*. For each walkable surface, we compute its free space stack, a set of boxes on top of the surface in which the *swept volume* of the robot $\mathcal{R}$ necessarily has to lie. We further represent the connectivity of surfaces by a graph structure. This graph structure then enables us identify homotopy classes.

We will consider a decomposition of the environment into a set of objects as

$$\mathbf{E} = O_1 \sqcup \cdots \sqcup O_\alpha \tag{1}$$

with O_i being a bounded convex polytope

$$O_i = \{x \in \mathbb{R}^3 | a_j^{(i)T} x \leq b_j^{(i)}, \|a_j^{(i)}\|_2 = 1, j \in [1, \alpha_i]\} \tag{2}$$

We make here the assumption that every object O_i is a convex polytope. If an object is not a convex polytope, we decompose it into convex subobjects [9], such that we can operate w.l.o.g. on convex polytopes.

For every object O_i, we define the p-th surface element as

$$\begin{aligned} S_i^p = \{x \in \mathbb{R}^3 | a_p^{(i)T} x = b_p^{(i)}, a_j^{(i)T} x \leq b_j^{(i)}, \\ j = 1, \cdots, p-1, p+1, \cdots, \alpha_i\} \end{aligned} \tag{3}$$

with $a_p^{(i)}$ the surface normal, and $b_p^{(i)}$ the distance to the origin.

Definition 1 (Walkable Surface)**.** *A surface element S_i^p is called* walkable*, if*

1) *the slope of S_i^p is smaller than the maximum slope $\mathcal{R}_\theta$ the robot can stand on*

$$\|a_p^{(i)} - v_g\| \leq \sqrt{(2 - 2\cos(\mathcal{R}_\theta))} \tag{4}$$

with $v_g = (0, 0, 1)^T$

2) *the foot of radius $\mathcal{R}_{FR}$ is fully contained inside S_i^p, meaning the following convex problem is feasible (based on the maximum inscribed circle problem [10])*

$$\begin{aligned} \underset{x \in \mathbb{R}^n, r \in \mathbb{R}}{\textit{maximize}} \quad & R \\ \textit{subject to} \quad & a_i^T x + R a_i^T a_i' \leq b_i, \\ & i = \{1, \cdots, p-1, p+1, \cdots, \alpha_i\} \\ & a_p^T x = b_p \\ & R \geq \mathcal{R}_{FR} \end{aligned} \tag{5}$$

whereby r is the radius of the circle, x the center, a_i' is the orthogonal projection onto the hyperplane of a_p, i.e. $a_i' = a_i - (a_i^T a_p) a_p$. Visualized in Fig. 3.

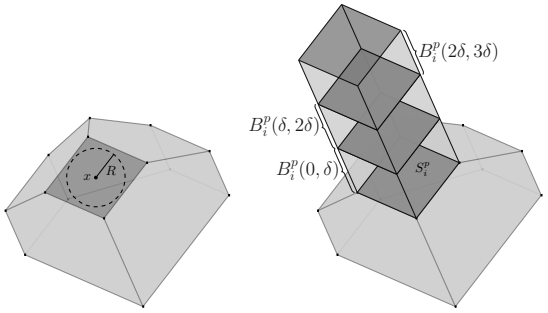


Fig. 3. **Left**: a polytope (light gray), a surface element (dark gray) and an inscribed circle with radius R and center x. **Right**: a set of cuboids B_i^p on top of one surface element S_i^p (dark gray)

We now add a notion of connectivity:

Definition 2 (Connectivity)**.** *Two walkable surfaces S_i, S_j are called* connected*, iff $d(S_i, S_j) < \mathcal{R}_{Step}$, with $\mathcal{R}_{Step}$ being the maximum step size of the robot $\mathcal{R}$*

This connectivity gives rise to a graph structure G_E, which contains walkable surfaces a nodes, and an edge between two connected surfaces.

To approximate the free space, we stack cuboids on top of each walkable surface. A cuboid of height δ and with distance Δ_L to S_i^p is defined as $B_i^p(\Delta_L, \Delta_L + \delta)$ See Fig. 3. The stack of cuboids on S_i^p will be denoted by

$$\begin{aligned} B_i^p &= \{B_{i,k}^p\}_{k=1}^{\beta} \\ B_{i,k}^p &= B_i^p(k\delta, (k+1)\delta)\} \end{aligned} \tag{6}$$

β is choosen such that $\beta > \frac{\mathcal{R}_{\mathrm{H_U}}}{\delta}$ with $\mathcal{R}_{\mathrm{H_U}}$ the maximum height of the robot. For each $B_{i,k}^p$, we apply a clipping algorithm [11] to decompose it into smaller convex cuboids.

Additionally we define the intersection element between two stacks of cuboids B_i, B_j as $I_{ij} = B_i \cap B_j$.

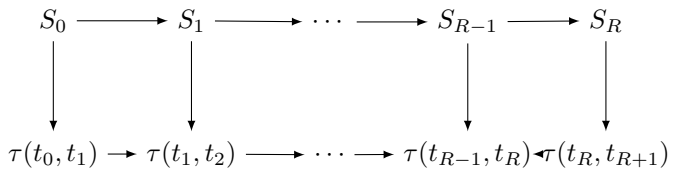


Fig. 4. A function τ has support on a walkable surface S_i at the time $[t_i, t_{i+1}]$.

Now, given two configurations $q_I, q_G \in \mathcal{C}$ of the robot, we compute the right foot position as $x_I = T(q_I), x_G = T(q_G)$ by using a forward kinematics operator T. Given x_I, x_G, we define $S_I = \underset{S_k \in S}{\operatorname{argmin}}\ d(x_I, S_k)$ to be the initial surface, and $S_G = \underset{S_k \in S}{\operatorname{argmin}}\ d(x_G, S_k)$ to be the goal surface.

Given S_I, S_G, we compute $\mathcal{H} = \{H_1, \cdots, H_R\}$, the set of R simple connected paths on the environment graph G_E. We call $H \in \mathcal{H}$ a homotopy, and we will write the connection of walkable surfaces as $H : S_0 \to \cdots \to S_{R_H}$. As a note, the complexity of finding all connected paths in a graph with V vertices is $\mathcal{O}(|V|!)$ [12].

To summarize, in this section we preprocessed the environment **E**, to decompose it into

- A set of N_w walkable surfaces $S_1, \cdots, S_{N_w}$
- A set of N_w stack of cuboids $B_1, \cdots, B_{N_w}$
- A set of N_i connector elements $I_1, \cdots, I_{N_i}$
- The environment graph G_E, describing the connectivity between walkable surfaces
- A set of homotopies $\mathcal{H}$ for given x_I, x_G

III. Convex optimization of Footpath Homotopies

Given $H : S_0 \to \cdots \to S_{R_H}$, our goal is to find a sliding footstep trajectory supported on the surfaces $S_0, \cdots, S_{R_H}$. More formally, we will consider the functional space of space curves as

$$\Omega(x_I, x_G, H) = C^1([0,1], \mathbb{R}^3) \tag{7}$$

under the constraints that for all $\tau \in \Omega(x_I, x_G, H)$ we have $\tau(0) = x_I$, $\tau(1) = x_G$, and that a segment $\tau(t)$ for $t \in [t_i, t_{i+1}]$ has support on a walkable surface S_i, as depicted in Fig. 4. In between support, we assume that the function is not supported, i.e. the foot can freely move through space, under the restriction that the non-support movement is smaller than the maximum stepsize.

We represent the functional space $\Omega(x_I, x_G, H)$ by a linear combination of basis functions [10], based on the Stone-Weierstrass theorem. The Stone-Weierstrass theorem [13] guarantees that every continuous function from $C([0,1], \mathbb{R}^3)$ can be approximated arbitrarily close by a polynomial function, i.e. we have

Theorem 1 (Stone-Weierstrass)**.** *Let $\tau \in C^1([0,1], \mathbb{R}^3)$. For every $\epsilon > 0$ there exists a polynomial $p(t) = \sum_{i=0}^{\infty} w_i t^i$ such that for all $t \in [0,1]$ we have*

$$\|\tau(t) - \sum_{i=0}^{\infty} w_i t^i\| < \epsilon \tag{8}$$

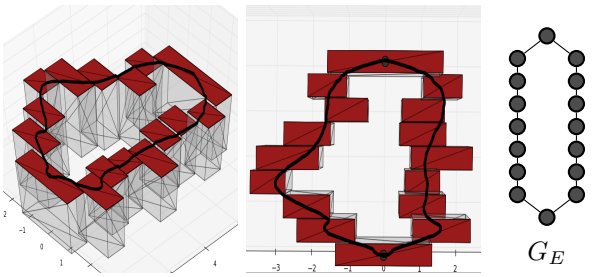


Fig. 5. Two homotopy classes of footsteps, and two solutions, obtained by solving one convex optimization program in each homotopy class. Also we show the environment graph G_E for this particular example, which represents connectivity between walkable surfaces.

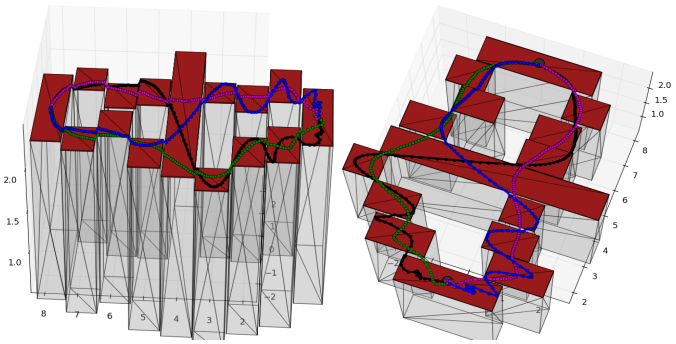


Fig. 6. Four homotopy classes in the environment: Our algorithm finds the homotopy classes via graph search, and then solves one convex optimization problem in each class to find the global optimal solution trajectory.

Conceptually, we represent a continuous function by a linear combination of an infinite set of basis functions. We will make the assumption that higher-order terms are negligible such that we choose a finite $K \gg 0$, and use $p(t) = \sum_{i=0}^{K-1} w_i t^i$. We will denote $F = \{x^0, \cdots, x^{K-1}\} \in \mathbb{R}^{K \times D}$, with K basis functions, and D the discretization of $[0,1]$. For all $t \in [0,1]$ we denote the approximation by $\tau = W^T F(t)$, with $W \in \mathbb{R}^{K \times 3}$. The complete convex optimization problem in homotopy class H becomes

$$\underset{\tau \in \Omega(x_I, x_G, H)}{\text{minimize}} \quad c(\tau) \tag{9}$$

$$\text{subject to} \quad \tau(0) = x_I, \tau(1) = x_G \tag{10}$$

$$\forall S_i \in \{S_0, \cdots, S_R\}, t \in [t_i, t_{i+1}] : \quad A_i \tau(t) \leq b_i \tag{11}$$

$$\forall t \in [0,1] : \quad \|\tau(t) - \tau(t + \Delta t)\| \leq \mathcal{R}_{\text{Step}} \tag{12}$$

$$\forall t \in [0,1] : \quad \|\tau(t) - W^T F(t)\| \leq \epsilon \tag{13}$$

whereby we have the following parameters

TABLE I

RESULTS FOR PLANNING PATHS IN THE TWO SCENARIOS SHOWN IN FIG. 5 AND FIG. 6. R IS THE NUMBER OF HOMOTOPIES, T_W IS THE TIME TO EXTRACT WALKABLE SURFACES FROM THE ENVIRONMENT, T_G THE TIME TO COMPUTE THE CONNECTIVITY BETWEEN SURFACES, T_P THE PLANNING TIME OF SOLVING R CONVEX OPTIMIZATION PROBLEMS, AND T IS THE ACCUMULATED TIME OF ALL STAGES TOGETHER (AVERAGED OVER 10 RUNS, ROUNDED).

Environment	R Homotopies	T_W(s)	T_G(s)	T_P(s)	T(s)
Stepping 1 (Fig. 5)	2	0.27	1.92	6.53	8.73
Stepping 2 (Fig. 6)	4	0.26	1.68	20.34	22.28

- $\mathcal{R}_{\text{Step}}$ the maximum step size of the robot
- $\epsilon > 0$ approximation constant to circumvent numerical instabilities
- K number of basis functions
- $c(\tau)$ is a convex objective function on τ, for example the shortest path as $c(\tau) = \int_{t=0}^{1} \|\tau(t) - \tau(t + \Delta t)\|_2^2 \text{dt}$

The given convex problem describes the set of all trajectories restricted to one homotopy class. A valid footstep at t can be modeled as a convex inequality constraint:

$$\forall t \in [0, 1] : A_i \tau(t) \leq b_i - \mathcal{R}_F \, \mathbf{diag}(A_i^T A_i') \tag{14}$$

whereby $A_i = \{a_0, \cdots, a_{M_i}\}$ contains the normals of the polytope associated to the walkable surface S_i, $A_i' = \{a_0', \cdots, a_{M_i}'\}$ with $a_j' = a_j - (a_j^T a_p) a_p$, and $\mathcal{R}_F$ being the radius of the foot. Compare to (5).

Fig. 5 shows the result of our convex optimization problem for an environment with 2 homotopy classes. A more complex version with 4 homotopy classes is shown in Fig. 6. The final planning results are depicted in Table I, all generated by using the splitting conic solver (SCS) [14] inside cvxpy [15].

IV. UPPER BODY OPTIMIZATION

We have showed so far how to optimize a single footstep trajectory $\tau_0 \in \Omega(x_I, x_G, H)$ in one homotopy class of the environment via a convex optimization problem. Now we assume that τ_0 is fixed. Our goal is to find a set of particle trajectories in the same homotopy class as τ_0, belonging to the swept volume of $\mathcal{R}$, such that those particles are feasible in $\mathbf{E}$. To put if differently, instead of searching for a single configuration space trajectory, we are searching for a set of mutually constrained particle trajectories in the environment. This section describes one possible way to constrain those particle trajectories to lie in the same homotopy class as τ_0. Please consult also Fig. 2 for an overview.

Each particle of the swept volume moves along a space curve in $\mathbb{R}^3$. Let $\Upsilon = \{\{\tau_k^l, \tau_k^r\}\}_{k=0}^{\eta}$ be the set of space curves of $2(\eta + 1)$ particles, with $\tau_k^{\{l,r\}} \in C^1([0, 1], \mathbb{R}^3)$, and τ_k^l represents the left outer hull of the swept volume of the robot at height $k\delta$, and τ_k^r the right hull. If we take a cross-section of the swept volume, then τ_k is represented by a point at height $k\delta$, as depicted by the red dots in Fig. 7. To achieve this, we apply three constraints on the functional space Υ

1) $\tau_k(t) \in P(t)$, the plane orthogonal to the foot trajectory τ_0 at instance t (cmp. Fig. 8)
$$P(t) = \{x \in \mathbb{R}^3 | a_P^T(t) x = b_P(t)\} \tag{15}$$
with $a_P(t) = \frac{\tau_0'(t)}{\|\tau_0(t)\|}$ and $b_P(t) = a^T(t)\tau_0(t)$.
2) $\tau_k(t)$ has to be feasible in $\mathbf{E}$, i.e. if τ_0 is supported on S_i^p at t, then
$$\tau_k(t) \in B_i^p(k\delta, (k + 1)\delta) \tag{16}$$
3) At instance t, all particles resemble a cross-section X_k of the robot
$$\tau_k \in X_k \tag{17}$$

The first two constraints are a linear equality and a convex inequality, respectively. The third constraint however is non-convex. To obtain X_k, we sample the configuration space $\mathcal{C}$ and compute cross-sections. A cross-section of a configuration is defined as its swept volume on the plane in movement direction, i.e. at t, the volume of $q \in \mathcal{C}$ is projected onto $P(t)$, as depicted in Fig. 7. As a simplification, we use only irreducible configurations of the robot [3]. An irreducible configuration is a configuration which has a minimal swept volume. Basically, we sample $\{q_1, \cdots, q_\sigma\} \in \mathcal{C}$, apply a cross-section operator $\phi : \mathcal{C} \to X \times X$ and compute the cross-sections $X = \{\{x_{l,1}, x_{r,1}\}, \cdots, \{x_{l,N}, x_{r,N}\}\}$. x_l stands for the left points of the swept volume, and x_r for the right points and we note that $x_r = A_l x_l$.

We now have to find a feasible cross-section for each plane. As a simplification, we consider the cross-sections only at intersections I_r of the environment, since those intersections represent the narrow passages. We note that we have only convex boxes in-between intersections, and so we assume that we can linearly interpolate two intersection points.

Our algorithm proceeds in the following manner: we compute the feasibility of N cross-sections by solving N convex optimization problems $\Theta_1^i, \cdots, \Theta_N^i$ for all intersections $i \in [1, V]$ in $I_1, \cdots, I_V$. A feasible path is then a sequence $\lambda_1, \cdots, \lambda_V$ of feasible cross-sections $\Theta_{\lambda_1}^1, \cdots, \Theta_{\lambda_V}^V$ with $\Theta_{\lambda_j}^i < \infty$. We can represent this as a solution matrix

$$\Lambda = \begin{bmatrix} \Theta_1^1 & \cdots & \Theta_1^V \\ \vdots & \ddots & \vdots \\ \Theta_N^1 & \cdots & \Theta_N^V \end{bmatrix} \tag{18}$$

whereby we have that Θ_j^i solves the problem of feasibility of a cross-section X_j on an intersection element I_i. Let t_i be such that $\tau(t_i) \in I_i$. Then Θ_j^i becomes

$$\Theta_j^i = \underset{\{\tau_0, \cdots, \tau_\eta\} \in \Upsilon}{\text{minimize}} \quad c(\tau_0, \cdots, \tau_\eta) \tag{19}$$

$$\text{subject to} \quad \forall k \in [0, \eta] :$$

$$\tau_k(t_i), A_L \tau_k(t_i) \in P(t_r) \tag{20}$$

$$\tau_k(t_i), A_L \tau_k(t_i) \in I_i(k\delta, (k + 1)\delta) \tag{21}$$

$$\tau_k(t_i), A_L \tau_k(t_i) = X_j \tag{22}$$

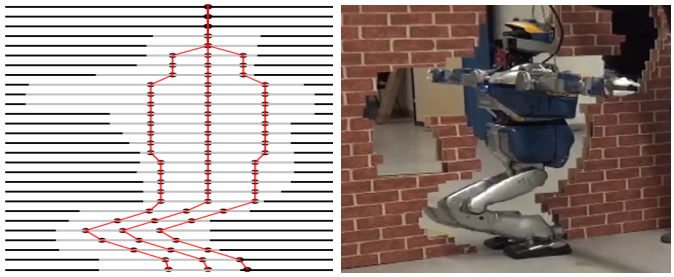


Fig. 7. **Left**: The cross-section space for a humanoid robot. Each line represents a certain height above a walkable surface. The cross-section of the robot intersects each height at two points, which we call x_L and x_R for left and right, respectively. We assume that every cross-section gives rise to only two points, i.e. we ignore configurations, where this is not the case. Overlaid (white line segments) are the constraints by the wall environment, which impose a convex box inequality on the cross-sections.**Right**: the final experiment, with the humanoid robot HRP-2 walking through a narrow environment.

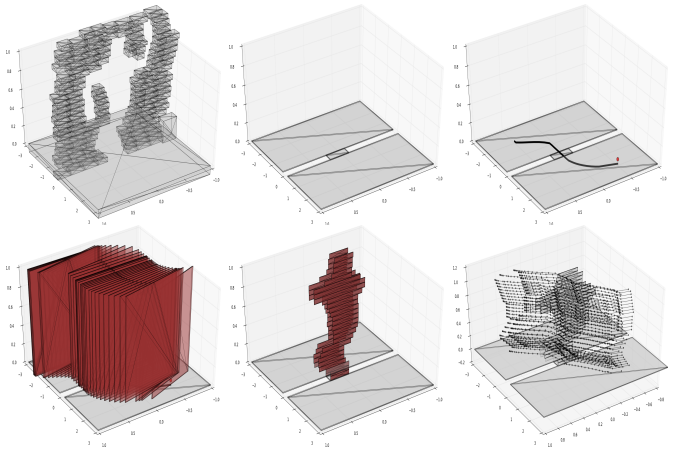


Fig. 8. From Left to Right, Top to Bottom: 1) all polytopes in the wall environment 2) the extracted walkable surfaces from our algorithm, 3) the footstep trajectory in the single homotopy class, computed by solving the convex optimization problem (9), 4) the vertical planes along the footstep trajectory, 5) the intersection of boxes on each walkable surface to create the connection elements 6) the final plan of workspace trajectories, all homotopically equivalent and representing a configuration space trajectory.

Fig. 8 represents the complete algorithmic output: from an environment, we compute walkable surfaces, we compute a footstep trajectory, we compute planes orthogonal to the footstep, we solve a set of convex optimization problems at each intersection, and we compute a final set of particle trajectories in the environment.

Finally, our main point here is that we have investigated the structure of the planning problem. Given our assumptions, we can compute the number of local minima of our planning problem as

$$L = \sum_{i=1}^{R} N^{V_i} \tag{23}$$

with R the number of homotopy classes, N the number of cross-sections, and V_i the number of intersections inside the i-th homotopy class. For the wall environment in Fig. 8 we have $N = 144$,$V = 2$,$R = 1$ and so $L = 20736$.

While our work is preliminary and non-complete, we want to stress the fact that knowing the number of local minima is important for understanding the inherent complexity of motion planning.

V. Experiments

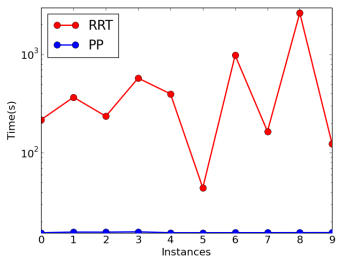
We implemented the algorithms in python, and used cvxpy [15] to compute solutions to the local minima. The source code to reproduced the experiments is available at

https://github.com/orthez/mpp-path-planner

For experimental verification, we have chosen the wall environment depicted in Fig. 8, which contains 145 objects.

The following parameters were used: $\beta = 40$, $\delta = 0.05$, such that $\beta\delta = 2.0 > \mathcal{R}_H$ with $\mathcal{R}_H = 1.539m$ the maximum height of HRP-2 [16]. For Problem (9), we used $\epsilon = 0.02$, $D = 1000$, $K = 2000$, and we used a minimum number of $D_i = 15$ samples for each walkable surface S_i.

The environment decomposition took $3m20s$, and our algorithm computed $N = 144$ cross-section configurations. We employ a greedy version of our algorithm, which computes



Planning Instance	RRT [3]	PP
0	3h37m	15m41s
1	6h07m	16m03s
2	3h55m	16m01s
3	9h34m	16m12s
4	6h37m	15m41s
5	0h44m	15m41s
6	16h19m	15m42s
7	2h45m	15m43s
8	43h57m	15m45s
9	2h03m	15m16s

Fig. 9. Comparison of running time on 10 instances for RRT (red) (adapted from [3]) and for our particle planning (PP) algorithm (blue). For PP, the time depends on the discretization

all local minima for the first intersection, and then checks if the next intersection can be solved by the solution to the first intersection. For each intersection, we computed all minima and found out that $11/144$ have been feasible (7.64%). The computation took $12m15s$ (averaged over 10 runs). All together we have a total computation time of $15m35s$. We compared the results of all runs with the results of a rapidly exploring random tree [6], operating on the irreducible configuration space [3]. The results in Fig. 9 show that we have a lower variance while performing better at the given sampling resolution.

To move the robot in the real world, we add small footsteps along the path, one every $0.1m$. Given footsteps and trajectory of the upper body, we use a dynamical solver to compute zero-moment point trajectories for the robot. We have used those results to verify the motion in the dynamical simulator OpenHRP [17], and executed it on the humanoid robot HRP-2 [16]. The video can be found here

http://homepages.laas.fr/aorthey/videos/wall-homotopy.mp4

VI. Related Work

Bhattacharya et al. [8] compute homotopy classes in the environment, and use them as a constraint for graph-based search. Our work is complementary in the sense that we investigate how to formulate planning in one homotopy class as a set of convex optimal problems, while their work investigates how to compute the homotopy classes in the first place.

The technique presented in [18] estimates a single homotopy class by growing random spheres. Our approach tries to be more systematic in that we reason about contact surfaces, and restrict the free space by the robots geometry. Also, we consider planning inside a homotopy class not as a potential field controller, but as a global optimization procedure.

The work by [19] consider sweeping a spherical object to find weakly collision free footstep positions. Our work is similar for footsteps but precomputes homotopy classes to identify high-level minima.

[20] identifies narrow passage in the environment, and computes important waypoint configurations inside those narrow passages. This idea inspired our computation of connector elements, elements which connect two contact surfaces.

Ivan et al. [21] introduce different topological representations to easier solve certain subproblems of motion planning. Our work is complementary, in that we would be able to analyze which representation to use given a certain problem.

The authors of [22] discover convex regions of footsteps in an environment, and employ mixed-integer programming to find a solution. Our work explores how adding more structure in form of connectivity can help to discover the homotopy classes, and formulate the resulting problem as a set of convex optimization problems.

Farber [23] introduced the topological complexity of a configuration space. Our work can be seen as a practical means of identifying the covering of the workspace volume and thereby its topological complexity. Our optimization algorithms are then one proposal to find paths inside of a given covering.

This work is fundamentally based on the work by [3], who introduced irreducible configuration for humanoid robots. Our work is complementary in that we are restricting our motions to the space of irreducible configuration while exploiting environment structure.

VII. Conclusion

We decomposed the general motion planning problem into a set of homotopic motion planning problems, with the goal of developing more efficient algorithms for the homotopic motion planning problem.

We presented three results: I) how to identify homotopy classes in an environment, based on walkable surfaces, surfaces on which a robot can make a foot contact. II) how to find a single contact trajectory inside a given homotopy class, formulated as a single convex optimization problem, and III) how to find a feasible upper body trajectory by solving a set of convex optimization problems.

Regarding future work, we currently work on incorporating our particle planning into a local motion planning algorithm to produce a dynamical feasible motion. We also would like to investigate when a surface is walkable, depending on physical properties like density, geometry, maximum pressure, and slippage. Finally, we would like to investigate the complexity properties of homotopic particle motion planning.

References

[1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Conference on Foundations of Computer Science*, pp. 421–427, 1979.

[3] A. Orthey, F. Lamiraux, and O. Stasse, "Motion Planning and Irreducible Trajectories," in *International Conference on Robotics and Automation*, 2015.

[4] A. El Khoury, F. Lamiraux, and M. Taix, "Optimal Motion Planning for Humanoid Robots," in *International Conference on Robotics and Automation*, 2013.

[5] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428 – 442, 2013.

[6] S. M. Lavalle and J. J. Kuffner Jr, "Rapidly-Exploring Random Trees: Progress and Prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000.

[7] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot path planning," in *Symposium on Theory of Computing*, pp. 353–362, ACM, 1995.

[8] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[9] J.-M. Lien and N. M. Amato, "Approximate convex decomposition of polyhedra and its applications," *Computer Aided Geometric Design*, vol. 25, no. 7, pp. 503–522, 2008.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[11] B. R. Vatti, "A Generic Solution to Polygon Clipping," *Communications of ACM*, vol. 35, pp. 56–63, July 1992.

[12] R. Sedgewick, "Part 5: Graph algorithms," in *Algorithms in C*, Addison Wesley Professional, 2001.

[13] W. Rudin, *Functional analysis*. International Series in Pure and Applied Mathematics, New York: McGraw-Hill Inc., second ed., 1991.

[14] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Operator splitting for conic optimization via homogeneous self-dual embedding," *arXiv preprint arXiv:1312.3039*, 2013.

[15] S. Diamond, E. Chu, and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization." http://cvxpy.org/, May 2014.

[16] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid robot HRP-2," in *International Conference on Robotics and Automation*, pp. 1083–1090, 2004.

[17] F. Kanehiro, H. Hirukawa, and S. Kajita, "OpenHRP: Open architecture humanoid robotics platform," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 155–165, 2004.

[18] O. Brock and L. E. Kavraki, "Decomposition-based Motion Planning: A Framework for Real-time Motion Planning in High-dimensional Configuration Spaces," in *International Conference on Robotics and Automation*, 2001.

[19] N. Perrin, O. Stasse, F. Lamiraux, and E. Yoshida, "Weakly collision-free paths for continuous humanoid footstep planning," in *International Conference on Intelligent Robots and Systems*, 2011.

[20] Y. Yang and O. Brock, "Efficient Motion Planning Based on Disassembly," in *Robotics: Science and Systems*, (Cambridge, USA), June 2005.

[21] V. Ivan, D. Zarubin, M. Toussaint, T. Komura, and S. Vijayakumar, "Topology-based representations for motion planning and generalization in dynamic environments with interactions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1151–1163, 2013.

[22] R. Deits and R. Tedrake, "Footstep Planning on Uneven Terrain with Mixed-Integer Convex Optimization," in *International Conference on Humanoid Robots*, 2014.

[23] M. Farber, "Topological complexity of motion planning," *Discrete and Computational Geometry*, vol. 29, no. 2, pp. 211–221, 2003.