Homotopic particle motion planning for humanoid robotics
Andreas Orthey, Vladimir Ivan, Maximilien Naveau, Yiming Yang, Olivier Stasse, Sethu Vijayakumar

To cite this version:
Andreas Orthey, Vladimir Ivan, Maximilien Naveau, Yiming Yang, Olivier Stasse, et al.. Homotopic particle motion planning for humanoid robotics. Rapport LAAS n° 15107. 2015. <hal-01137918>

HAL Id: hal-01137918
https://hal.archives-ouvertes.fr/hal-01137918
Submitted on 31 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Homotopic particle motion planning for humanoid robotics

Andreas Orthey¹, Vladimir Ivan², Maximilien Naveau¹, Yiming Yang², Olivier Stasse¹, Sethu Vijayakumar²

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 elderly care, deep space exploration, and nuclear waste removal.

We study here the motion planning problem $A = \{\mathcal{R}, \mathcal{C}, q_I, q_G, \mathcal{E}\}$ with $\mathcal{R}$ 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 $\mathcal{E}$ 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 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 $\{\tau_k\}_k=0$ on the robot body moving through $\mathbb{R}^3$ on space curves of the form $\tau_k : [0, 1] \rightarrow \mathbb{R}^3$. 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.

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 constrained by the contact trajectory, as a set of convex optimization problems

Sec. II describes how we decompose an environment into walkable surfaces, homotopy classes and the free space inside of one homotopy class. Sec. III formulates optimizing
Fig. 2. The conceptual overview about this paper. Top: in section II we decompose the environment E into walkable surfaces S, intersections I and intersections T. From the start contact x1, the goal contact xG and the connectivity graph GE we compute the homotopy classes H on S. Middle: Given the homotopy classes H, the start contact x1 and the goal contact xG, we compute a sliding footstep trajectory τ0 supported on H. We compute upper-body particle trajectories Y from a given footstep trajectory τ0, planes P, cuboids B, and from cross-sections X generated by sampling robot configurations 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 referred 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 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

\[ E = O_1 \sqcup \cdots \sqcup O_{\alpha} \]  

with \( O_i \) being a bounded convex polytope

\[ O_i = \{ x \in \mathbb{R}^3 | a^{(i)}_p x \leq b^{(i)}_j, \| a^{(i)}_p \|_2 = 1, j \in [1, \alpha_i] \} \]  

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

\[ S^p_i = \{ x \in \mathbb{R}^3 | a^{(i)}_p x = b^{(i)}_j, a^{(i)}_p x \leq b^{(i)}_j \}, \]

\[ j = 1, \cdots, p - 1, p + 1, \cdots, \alpha_i \]  

with \( a^{(i)}_p \) the surface normal, and \( b^{(i)}_j \) the distance to the origin.

Definition 1 (Walkable Surface). A surface element \( S^p_i \) is called walkable, if

1) the slope of \( S^p_i \) is smaller than the maximum slope \( \mathcal{R}_\theta \) the robot can stand on

\[ \| a^{(i)}_p - v_g \| \leq \sqrt{2 - 2 \cos(\mathcal{R}_\theta)} \]  

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

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

\[
\begin{align*}
\text{maximize} & \quad R \\
\text{subject to} & \quad a^{(i)}_i x + Ra^{(i)}_a a'_i \leq b_i, \\
& \quad i = \{1, \cdots, p - 1, p + 1, \cdots, \alpha_i\} \\
& \quad a^{(i)}_p x = b_p \\
& \quad R \geq \mathcal{R}_{FR}
\end{align*}
\]

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. \( a'_i = a_i - (a^T_i a_p) a_p \). 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 \( B^p_i \) on top of one surface element \( S^p_i \) (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 as 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 \( \delta \) and with distance \( \Delta_L \) to \( S^p_i \) is defined as \( B^p_i(\Delta_L, \Delta_L + \delta) \) See Fig. 3. The stack of cuboids on \( S^p_i \) will be denoted by

\[
\begin{align*}
B^p_i &= \{ B^p_{i,k}\}_{k=1}^\beta \\
B^p_{i,k} &= B^p_i(k\delta, (k + 1)\delta)
\end{align*}
\]

\( \beta \) is choosen such that \( \beta > \frac{\mathcal{R}_{Hi}}{\mathcal{R}_{FR}} \) with \( \mathcal{R}_{Hi} \) the maximum height of the robot. For each \( B^p_{i,k} \), we apply the 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 \( I_{ij} = B_i \cap B_j \).
Now, given two configurations \( q_1, q_G \in C \) of the robot, we compute the right foot position as \( x_I = T(q_1), x_G = T(q_G) \) by using a forward kinematics operator \( T \). Given \( x_I, x_G \), we define \( S_I = \text{argmin}_{S_k \in S} d(x_I, S_k) \) to be the initial surface, and \( S_G = \text{argmin}_{S_k \in S} d(x_G, S_k) \) 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 \( H : S_0 \rightarrow \cdots \rightarrow S_{R_H} \). As a note, the complexity of finding all connected paths in a graph with \( V \) vertices is \( O(|V|!) \) [12].

To summarize, in this section we preprocess 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 \( H : S_0 \rightarrow \cdots \rightarrow S_{R_H} \), 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

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

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

\[
\|\tau(t) - \sum_{i=0}^{\infty} w_i t^i\| < \epsilon
\]

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 \( F = \{ x^0, \cdots, x^K \} \in \mathbb{R}^{K \times D} \), 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

\[
\text{minimize}_{\tau \in \Omega(x_I, x_G, H)} c(\tau) \\
\text{subject to} \\
\tau(0) = x_I, \tau(1) = x_G \\
\forall S_i \in \{ S_0, \cdots, S_{R_H} \}, t \in [t_i, t_{i+1}] : \\
A_i \tau(t) \leq b_i \\
\forall t \in [0, 1] : \\
\|\tau(t) - \tau(t + \Delta t)\| \leq R_{\text{Step}} \\
\forall t \in [0, 1] : \\
\|\tau(t) - W^T F(t)\| \leq \epsilon
\]

whereby we have the following parameters
The trajectories in the same homotopy class as \( \tau_0 \) at instance \( t \) (cmp. Fig. 5)

\[
P(t) = \{ x \in \mathbb{R}^3 | a_F^T(t)x = b_F(t) \} \tag{15}
\]

de the sweeping conic solver (SCS) [14] inside cvxpy [15].

\[a_F(t) = \frac{\tau_0(t)}{||\tau_0(t)||} \quad \text{and} \quad b_F(t) = a_T(t)\tau_0(t).\] \( \tag{16} \)

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 \( C \) 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 \( q \in C \) is projected onto \( P(t) \), as depicted in Fig. 7. As a simplification, we use only irreducible configurations of the robot. An irreducible configuration is a configuration which has a minimal swept volume. Basically, we sample \( \{q_1, \ldots, q_{\tau}\} \in C \), apply a cross-section operator \( \phi : C \to X \times X \) and compute the cross-sections \( X = \{ \{x_{i,1}, x_{i,1}\}, \ldots, \{x_{i,N}, x_{i,N}\}\} \). \( x_i \) stands for the left points of the swept volume, and \( x_r \) for the right points and we note that \( x_r = A_i x_l \).

We now have to find a feasible cross-section for each plane. As a simplification, we consider the cross-sections only at intersections \( I_i \) 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 \( \Theta_{i1}, \ldots, \Theta_{iN} \) for all intersections \( i \in [1, V] \) in \( I_1, \ldots, I_V \). A feasible path is then a sequence \( \lambda_1, \ldots, \lambda_V \) of feasible cross-sections \( \Theta_{i\lambda_1}, \ldots, \Theta_{i\lambda_V} \) with \( \Theta_{i\lambda_0} < \infty \). We can represent this as a solution matrix

\[
\Lambda = \begin{bmatrix}
\Theta_{i1} & \cdots & \Theta_{iN} \\
\vdots & \ddots & \vdots \\
\Theta_{i1} & \cdots & \Theta_{iN}
\end{bmatrix} \tag{18}
\]

whereby we have that \( \Theta_{ij} \) solves the problem of feasibility of a cross-section \( X_j \) on an intersection element \( I_i \). Let \( t_i \) be such that \( \tau_i(t) \in I_i \), then \( \Theta_{ij} \) becomes

\[
\Theta_{ij} = \minimize_{\{\tau_0, \ldots, \tau_{\eta}\}} c(\tau_0, \ldots, \tau_{\eta}) \tag{19}
\]

subject to \( \forall k \in [0, \eta] : \)

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

\[
\tau_k(t_i), A_L\tau_k(t_i) \in I_k(k\delta, (k+1)\delta) \tag{21}
\]

\[
\tau_k(t_i), A_L\tau_k(t_i) = X_j \tag{22}
\]
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.

For Problem (9), we used \( \epsilon = 0 \) 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 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

\[
L = \sum_{i=1}^{R} N^{V_i}
\]

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 \( \beta \delta = 2.0 > R_H \) with \( R_H = 1.539m \) 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 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/orthez/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