

---

# Path Planning on Manifolds using Randomized Higher-Dimensional Continuation

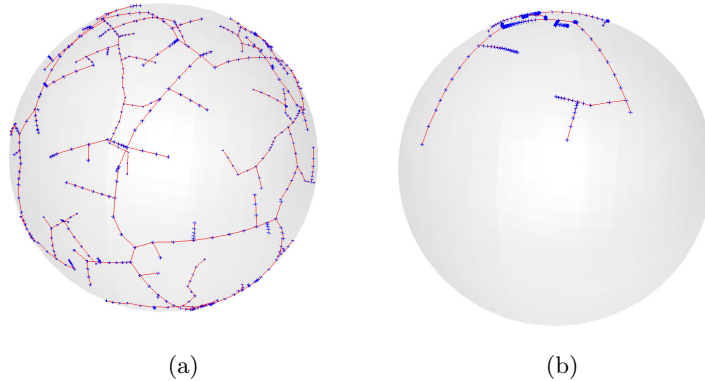
Josep M. Porta and Léonard Jaillet

Institut de Robòtica i Informàtica Industrial, CSIC-UPC  
{porta,ljaillet}@iri.upc.edu

**Abstract:** Despite the significant advances in path planning methods, problems involving highly constrained spaces are still challenging. In particular, in many situations the configuration space is a non-parametrizable variety implicitly defined by constraints, which complicates the successful generalization of sampling-based path planners. In this paper, we present a new path planning algorithm specially tailored for highly constrained systems. It builds on recently developed tools for Higher-dimensional Continuation, which provide numerical procedures to describe an implicitly defined variety using a set of local charts. We propose to extend these methods to obtain an efficient path planner on varieties, handling highly constrained problems. The advantage of this planner comes from that it directly operates into the configuration space and not into the higher-dimensional ambient space, as most of the existing methods do.

## 1 Introduction

Many problems require to determine a path between two points, fulfilling a given set of constraints. In Robotics, this appears for instance in parallel manipulators [33], robot grasping [23], constraint-based object positioning [22], surgery robots [1], and humanoid robots [18]. This situation also appears in Biochemistry when searching for low energy paths between different molecular conformations [37]. In all these cases, the constraints expressed as a set of equations reduce the configuration space to a variety composed by one or more manifolds embedded in a higher-dimensional ambient space, defined by the variables involved in the equations. Approaches that try to directly describe these manifolds exist, but they are either too complex to be applied in practice [4], or limited to particular architectures [26]. The adaptation of sampling-based planning methods is also cumbersome since, sampling in the ambient space, the probability of the samples to lay on the configuration space is null. Consequently, several methods have been devised to find points of the configuration space from points of the ambient space.

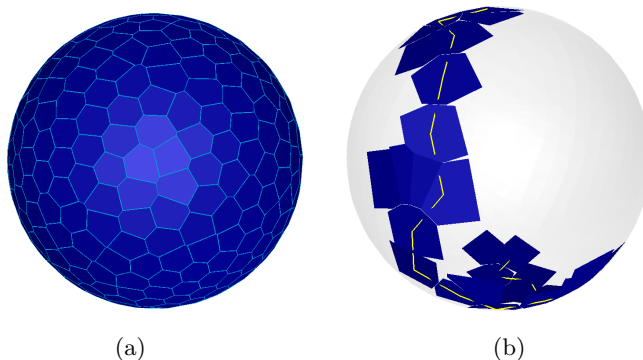


**Fig. 1.** RRTs with 500 samples. Blue crosses represent the tree nodes and red lines the connections between them. (a) When the ambient space is a box tightly enveloping the sphere, the exploration is relatively homogeneous. (b) When the box is elongated along the vertical axis, an unwanted bias penalize the exploration.

Kinematics-PRM [10] samples a subset of variables and uses inverse kinematics to find all the possible values for the remaining ones. This strategy is only valid for particular mechanisms, and although some improvements have been proposed [6], the probability of generating invalid samples is significant. Moreover, the presence of singularities in the subset of variables solved via inverse kinematics complicates the approach [9].

An alternative strategy to get a valid configuration is to use numerical iterative techniques, either implementing random walks [38], or the more efficient Jacobian pseudo inverse method [2, 7, 28]. All these approaches only perform properly when the ambient and the configuration spaces are similar. If the constraints define one or several complex surfaces with many folds, a uniform distribution of samples in the ambient space will not translate to a uniform distribution in the configuration space and this heavily reduces the efficiency of the sampling approaches. This problem may appear even in simple cases such as the one described in Fig. 1, where a RRT is build on a sphere from points sampled in a 3D ambient space. If the sphere is not centered in, and tightly enveloped by the ambient space, the sampling process is biased and the result is a poor exploration of the solution variety. The lack of prior knowledge on the variety structure makes it hard to forecast whether or not a sampling-based approach would be successful.

One way to limit the problems of mismatching between the two spaces is to focus the sampling in a subset of the ambient space around the configuration space [41]. However, even in the case where the ambient and the configuration spaces are somehow similar, samples are thrown in the ambient space that can be of much higher dimensionality than the configuration space. Ideally, one



**Fig. 2.** (a) Atlas of the sphere obtained by Higher-dimensional Continuation. Each polytope is a chart that locally parametrizes the sphere. The atlas includes about 500 charts. (b) Part of the atlas explored with our approach for connecting the two poles. Only about 30 charts are generated. The solution path is shown as a yellow line.

would like to sample directly on the lower-dimensional configuration space. A uniform sampling over the configuration space typically relies on a global parametrization of this space. In some families of mechanism distance-based formulations provide this parametrization [11, 29], some approaches try to infer it from large sets of samples [12], and task-space planners assume that a subset of variables related with the end-effector are enough to parametrize the configuration space [39, 25]. However, it is in general not possible to obtain a global isometric parametrization of the configuration space.

From differential geometry, it is well known that a variety can be described by an atlas containing a collection of charts, each chart providing a local parametrization of the variety [20]. In this context, Um *et al* [34] sketch a RRT scheme where the exploration is done in the tangent space and the samples are projected to the configuration space using the Jacobian pseudo inverse. However, the overlap between charts can lead to an inappropriate sampling bias and the potential presence of singularities and branching inside the variety is not considered. Higher-dimensional Continuation techniques provide principled numerical tools to compute the atlas of an implicitly defined variety departing from a point, avoiding overlap between neighboring charts [15]. For instance, Fig. 2(a) shows the atlas obtained with one of these methods [13] in the toy problem of the sphere. One-dimensional continuation methods (also known as path following, homotopy or bootstrap methods), have been strongly developed in the context of Dynamical Systems [17], whereas in Robotics, they have been mainly used for solving problems related to Kinematics [24, 27]. To our knowledge, Higher-dimensional Continuation tools have not been used in Robotics.

In this paper, we extend the tools developed for Higher-dimensional Continuation to the context of path planning. We define the concept of partial atlas connecting two configurations, dealing with the presence of obstacles. We also introduce the random exploration of a variety focused towards a target configuration. As a result, we obtain a Higher-dimensional Continuation planner (HC-planner) for highly constrained systems that clearly outperforms existing approaches. Fig. 2(b) shows an example of path found with our approach for the sphere toy problem. Note how a small subset of all the atlas charts is needed to find a path connecting the two given samples.

Next section provides a description of the tools for High-dimensional Continuation. Section 3 presents an extension these tools to the context of path planning. Section 4 compares the performance of the planner with respect to existing methods for several benchmarks. Finally, Section 5 summarizes the contributions of this work and indicates points that deserve further attention.

## 2 Higher-dimensional Continuation

Next, we describe the main algorithmic tools appearing in [13]. By generalizing the one-dimensional pseudo-arclength procedure, these tools allow the generation of an atlas for describing a  $k$ -dimensional variety implicitly defined by a system of equations

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

with  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-k}$ ,  $n > k > 0$ . Fig 3 illustrates the main idea on which relies the approach. Given a point  $\mathbf{x}_i$  on the variety, we can define  $\Phi_i$ , an orthonormal basis of the tangent space of the variety at this point. This is the  $n \times k$  matrix satisfying

$$\begin{pmatrix} \mathbf{J}(\mathbf{x}_i) \\ \Phi_i^\top \end{pmatrix} \Phi_i = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix}, \quad (2)$$

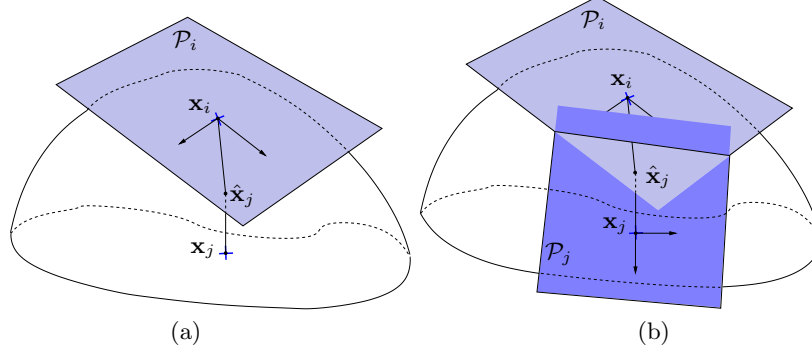
with  $\mathbf{J}(\mathbf{x}_i)$  the Jacobian of  $\mathbf{F}$  evaluated at  $\mathbf{x}_i$  and  $\mathbf{I}$  the identity matrix. The pair  $(\mathbf{x}_i, \Phi_i)$  defines a chart,  $\mathcal{C}_i$ , that locally approximates the variety. The root point,  $\mathbf{x}_j$ , of a new chart,  $\mathcal{C}_j$ , can be obtained by first generating a point,  $\hat{\mathbf{x}}_j$ , on the tangent space of  $\mathcal{C}_i$

$$\hat{\mathbf{x}}_j = \mathbf{x}_i + \Phi_i \mathbf{u}_i^j, \quad (3)$$

with  $\mathbf{u}_i^j$  a  $k$ -dimensional parameter vector. Then  $\mathbf{x}_j$  is the orthogonal projection of  $\hat{\mathbf{x}}_j$  into the variety. This projection is obtained by solving the system [21]

$$\begin{aligned} \mathbf{F}(\mathbf{x}_j) &= \mathbf{0}, \\ \Phi_i^\top (\mathbf{x}_j - \hat{\mathbf{x}}_j) &= \mathbf{0}, \end{aligned} \quad (4)$$

using a Newton procedure where  $\mathbf{x}_j$  is initialized to  $\hat{\mathbf{x}}_j$  and where at each iteration  $\mathbf{x}_j$  is updated with the increment  $\Delta \mathbf{x}_j$  fulfilling



**Fig. 3.** Higher-dimensional Continuation method applied to a two-dimensional manifold. (a) A chart is defined from the tangent space at a given start point  $\mathbf{x}_i$ . The area of applicability of the chart is denoted as  $\mathcal{P}_i$ . A point  $\hat{\mathbf{x}}_j$  from the tangent space at  $\mathbf{x}_i$  is orthogonally projected to the manifold to determine  $\mathbf{x}_j$ , the root point of the next chart. (b) The new chart locally parametrizes a new region of the manifold. The area of applicability of the new chart is  $\mathcal{P}_j$  that does not overlap with  $\mathcal{P}_i$ .

$$\begin{pmatrix} \mathbf{J}(\mathbf{x}_i) \\ \Phi_i^\top \end{pmatrix} \Delta \mathbf{x}_j = - \begin{pmatrix} \mathbf{F}(\mathbf{x}_i) \\ \Phi_i^\top (\mathbf{x}_j - \hat{\mathbf{x}}_j) \end{pmatrix}. \quad (5)$$

This is repeated until the norm of the right-hand side of the previous system becomes negligible or for a maximum number of iterations. When a valid  $\mathbf{x}_j$  is determined, we can define a new chart  $\mathcal{C}_j$  (Fig. 3(b)). The intersection between tangent spaces marks the boundaries of applicability of the corresponding charts, denoted as  $\mathcal{P}_i$  and  $\mathcal{P}_j$ , respectively. When  $\mathcal{C}_i$  is fully surrounded by other charts,  $\mathcal{P}_i$  becomes a convex polytope.

The algorithm proposed in [13] gives a systematic way to define new charts and to generate the associated polytopes. In this work,  $\mathcal{P}_i$  is initialized as an hypercube enclosing a ball,  $\mathcal{B}_i$ , of radius  $r$  (see Fig. 4(a)). The polytope is represented using a set of faces that intersect defining a set of vertices [5]. A vertex  $\mathbf{v}$  of  $\mathcal{P}_i$  external to  $\mathcal{B}_i$ , can then be used to generate a new chart. From this vertex, a point  $\hat{\mathbf{x}}_j$  on the surface of  $\mathcal{B}_i$  is defined using Eq. 3 and

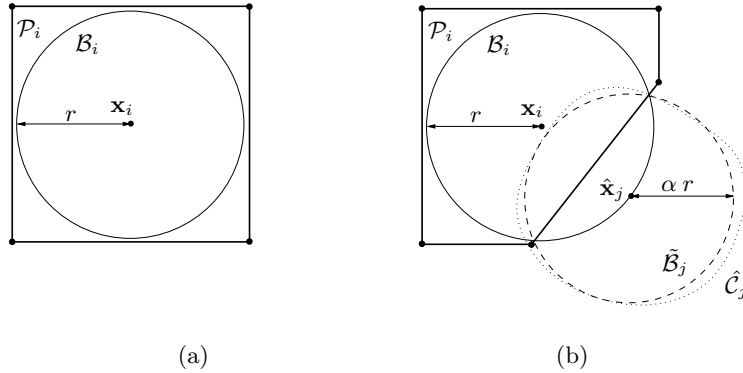
$$\mathbf{u}_i^j = \frac{r}{\|\mathbf{v}\|} \mathbf{v}. \quad (6)$$

If  $\mathcal{C}_i$  and the new  $\mathcal{C}_j$  generated from  $\mathbf{u}_i^j$  are too far or too different, i.e., if

$$\|\mathbf{x}_j - \hat{\mathbf{x}}_j\| > \sigma, \quad (7)$$

$$\|\Phi_i^\top \Phi_j\| < 1 - \sigma, \quad (8)$$

the new chart is discarded and a new attempt of chart generation is performed from a set of parameters  $\mathbf{u}_i^j$  closer to  $\mathbf{x}_i$ . When  $\mathcal{C}_j$  is valid, it is used to refine  $\mathcal{P}_i$  from the intersection between  $\mathcal{B}_i$  and  $\hat{\mathcal{C}}_j$ , the projection into the tangent space



**Fig. 4.** (a) Polytope-based chart construction by higher continuation methods. The initial polytope is a box including a ball of radius  $r$  around  $x_i$ . (b) The polytope is progressively refined using balls (dashed line) that approximate the projection of balls from nearby charts into the tangent space of the chart of interest (dotted line).

of  $\mathcal{C}_i$  of the part of the variety covered by  $\mathcal{C}_j$ . This projection is approximated by a ball,  $\tilde{\mathcal{B}}_j$ , included in  $\hat{\mathcal{C}}_j$ , as shown in Fig. 4(b). For this approximation to be conservative, the radius of  $\tilde{\mathcal{B}}_j$  is scaled by a factor  $\alpha$ ,  $0 < \alpha < 1$ , depending on the angle between sub-spaces spanned by  $\Phi_i$  and  $\Phi_j$ .

The hyperplane defined by the intersection of  $\mathcal{B}_i$  and  $\tilde{\mathcal{B}}_j$  can be computed by subtracting the equations for the two balls. As shown in Fig. 4(b), this plane defines a new face of  $\mathcal{P}_i$  that eliminates some of the vertices of  $\mathcal{P}_i$  (in particular the one used to generate  $\mathcal{B}_j$ ) and generates new vertices.  $\mathcal{P}_j$ , the polytope associated to  $\mathcal{B}_j$ , is cropped using the projection of  $\mathcal{C}_i$  into  $\mathcal{C}_j$ .

When all the vertices of the polytope of a chart are inside the associated ball, the chart cannot be further expanded as the domain for this chart is fully bounded. This process of chart expansion continues as far as there are open charts. At the end, the final atlas is made of charts fully recovering the variety, whose area of validity is bounded by the corresponding polytopes (see Fig. 2(a)). To fully characterize the variety, higher-dimensional continuation tools also need to consider singular points of the variety. This is addressed in [14], but details are omitted here for conciseness.

The cost of the algorithm at each step is dominated by the cost of two searches among the set of charts: one to find an open chart and another to find the potential neighbors of a new chart. The first search can be saved keeping the open charts in a list. The performance of the second search can be increased using a kd-tree storing the root points of the charts.

### 3 Path-Planning on Manifolds

Using the tools described in previous section, it is possible to characterize the connected component of a variety containing a starting point. A graph can be build where nodes are the charts and edges represent their neighboring relations. Thus, the shortest path in the graph to a given sample can be easily computed using a standard graph search method such as A\* considering only the collision free transitions between the chart roots. This procedure defines an optimal, resolution complete path planner, but it is only practical for low dimensional varieties, specially if we have to use small charts.

If we define charts with large radius, the presence of obstacles becomes an issue. The procedure described in Section 2 can deal with boundaries in the configuration space, but obstacles are not defined in this space. Moreover, the procedure used to extend charts is based on selecting a finite set of points that are the vertices of the charts polytopes and the presence of obstacles can block those predefined expansion directions.

Herein, we propose modifications to the Higher-order Continuation procedures to deal with the curse of dimensionality and the presence of obstacles. First, we take advantage of that path planning is only concerned with the path between two given configurations and not with the full atlas generation, which allows to save the construction of many unnecessary charts. Second, to deal with the presence of obstacles, we randomize the process of atlas extension.

#### 3.1 Chart selection: focusing in the path to the goal

As aforementioned, the atlas structure can be represented by a graph where nodes are the charts and edges are the neighboring relations between charts. To guide the search toward the goal, we use a Greedy Best-First search where the chart to expand is the one with minimum expected cost to reach the goal. The cost for a chart  $\mathcal{C}_i$  is heuristically evaluated as

$$h(i) = \beta^{n_i} \|\mathbf{x}_i - \mathbf{x}_g\|, \quad (9)$$

where  $\mathbf{x}_g$  is the goal configuration,  $\beta > 1$  a fixed parameter, and  $n_i$  is the number of times a chart failed to expand. Thus, the term  $\beta^{n_i}$  prevents the search to get stuck in dead ends. As soon as the goal is connected to the rest of the atlas, the search is stopped.

Observe that using a Greedy Best-First search, we do not necessary generate all the neighbors of the chart under expansion. The generation of children charts proceeds only while the children have higher cost than the parent. This largely reduces the generation of charts.

Finally, note that due to the use of a Greedy Best-First algorithm the final path is not necessarily optimal. The generation of a (resolution) optimal path would require the use of an A\* algorithm and the generation of all the neighbors of the node/chart under expansion. In general, this implies to generate too many charts, hindering the practical applicability of the approach.

### 3.2 Chart expansion: Generating random directions

When the chart to be expanded is selected, the expansion point for the atlas is selected at random. This is achieved by sampling a point uniformly on the surface of the ball associated with the atlas and checking if this point is inside the associated polytope. If it is the case, the generation of the new atlas proceeds as detailed in Section 2.

The uniform generation of random point  $\mathbf{u}_i^j$  on a the surface of a  $k$ -dimensional ball is done by generating the point elements according to a normalized one-dimensional Gaussian and scaling the resulting vector to norm  $r$  [8].

To verify if the point is inside the associated chart’s polytope, we exploit the fact that polytopes are defined as the intersection of  $k$ -dimensional hyperplanes. Thus, for a point  $\mathbf{u}_i^j = \{u_1, \dots, u_k\}$  to be inside the polytope  $\mathcal{P}_i$  made of  $m_i$  faces, it must fulfill

$$\gamma_0^t + \sum_{s=1}^k \gamma_s^t u_s \geq 0, \quad (10)$$

for all the faces  $f^t = (\gamma_0^t, \dots, \gamma_k^t)$ ,  $t = 1, \dots, m_i$  defining  $\mathcal{P}_i$ .

If the point is inside the polytope, it is approached through small incremental steps of size  $\delta$ . The intermediate points are successively projected on the manifold (using Eqs. 4 and 5) and then checked for collision. If no progress can be done towards the target point, the expansion is declared as failure and the chart under expansion is penalized by increasing  $n_i$ .

Observe that while the chart to extend is selected greedily, the exact expanding direction is selected randomly, favoring the exploration of alternative paths in the presence of obstacles.

### 3.3 Algorithm

Algorithm 1 corresponds to the HC-Planner, implementing the path planning approach introduced in this paper. The algorithm takes  $\mathbf{x}_s$  and  $\mathbf{x}_g$  as start and a goal configurations respectively, and tries to connect them with a path on the variety implicitly defined by a given set of constraints  $\mathbf{F}$ , as expressed in Eq. 1. The process begins by initializing two charts associated to the two query configurations (lines 1-2). Each chart includes the initial point  $\mathbf{x}$ , the base of the tangent space  $\Phi$ , the ball  $\mathcal{B}$  and the polytope  $\mathcal{P}$  limiting the area of influence of the chart. The two charts are then included in the initial atlas (line 3). To efficiently determine the chart with the minimum expected cost, charts are organized into a binary heap. Thus, the cost-to-goal of the start configuration is evaluated (line 4) and used to initialize the heap (line 5). In lines 6 to 16, a greedy search is performed as described in Section 3.1, while the two query configurations are disconnected. At each iteration, we extract  $\mathcal{C}_i$ , the most promising chart from the heap (line 7) and if the polytope  $\mathcal{P}_i$



**Algorithm 1:** High-dimensional Continuation path planner.

---

```

HC-Planer( $\mathbf{x}_s, \mathbf{x}_g, \mathbf{F}$ )
  input : A couple of samples to connect  $\mathbf{x}_s, \mathbf{x}_g$ , and a set of constraints  $\mathbf{F}$ .
  output: A path connecting the two samples
  1  $\mathcal{C}_s \leftarrow \text{INITCHART}(\mathbf{x}_s, \mathbf{F})$  //  $\mathcal{C}_s = \{\mathbf{x}_s, \Phi_s, \mathcal{B}_s, \mathcal{P}_s\}$ 
  2  $\mathcal{C}_g \leftarrow \text{INITCHART}(\mathbf{x}_g, \mathbf{F})$  //  $\mathcal{C}_g = \{\mathbf{x}_g, \Phi_g, \mathcal{B}_g, \mathcal{P}_g\}$ 
  3  $A \leftarrow \{\mathcal{C}_s, \mathcal{C}_g\}$ 
  4  $h(s) \leftarrow \|\mathbf{x}_s - \mathbf{x}_g\|$ 
  5  $H \leftarrow \text{INITHEAP}(\mathcal{C}_s, h(s))$ 
  6 while not CONNECTED( $A, \mathcal{C}_s, \mathcal{C}_g$ ) do
  7    $\mathcal{C}_i \leftarrow \text{EXTRACTMIN}(H)$  //  $\mathcal{C}_i = \{\mathbf{x}_i, \Phi_i, \mathcal{B}_i, \mathcal{P}_i\}$ 
  8   if  $\mathcal{P}_i \not\subseteq \mathcal{B}_i$  then
  9      $\mathcal{C}_j \leftarrow \text{GENERATENEWCHART}(\mathcal{C}_i, \mathbf{F})$ 
 10     if  $\mathcal{C}_j = \emptyset$  then
 11        $h(i) \leftarrow \beta h(i)$ 
 12        $H \leftarrow \text{ADDTOHEAP}(\mathcal{C}_i, h(i))$ 
 13     else
 14        $A \leftarrow A \cup \{\mathcal{C}_j\} \cup \text{SINGULARCHART}(\mathcal{C}_i, \mathcal{C}_j)$ 
 15        $h(i) \leftarrow \|\mathbf{x}_j - \mathbf{x}_g\|$ 
 16        $H \leftarrow \text{ADDTOHEAP}(\mathcal{C}_j, h(j))$ 
 17 RETURN(PATH( $A, \mathcal{C}_s, \mathcal{C}_g$ ))

```

---

of this chart still has vertices outside the ball  $\mathcal{B}_i$ , we try to extend the atlas with a new chart (line 9). If the extension fails (line 10), the current chart is penalized so that its chance to be selected for future extension decreases (line 11), and the chart is added to the heap with the updated cost (line 12). If the atlas extension succeeds, the new chart is added to the atlas, updating the neighboring relations between charts (line 14). Next, the heuristic-to-goal is initialized for the new chart (line 15) and added to the atlas (line 16). When the goal is reached, a graph search procedure can be used to extract the path linking the query configurations. Every time a new chart is added to the atlas, we check whether the line connecting the roots of the parent and the child charts crosses a singularity. If so, the singular point is located and a chart is defined such that its root is at the singularity and its tangent space is aligned with the branch of the variety that does not contain  $\mathcal{C}_i$  and  $\mathcal{C}_j$ . Function SINGULARCHART (line 14) returns this chart or an empty set if there is no singularity between  $\mathcal{C}_i$  and  $\mathcal{C}_j$ .

The generation of a new chart from a previous one is presented in Algorithm 2. We select a point  $\mathbf{u}_i^j$  on the ball defined on the tangent space of the input chart as described in Section 3.2 (line 2). If the point is inside the polytope, the point is not in the area of influence of a neighboring chart, and thus, we proceed to determine a point,  $\mathbf{x}_j$ , adequate to generate a new chart. This point is searched from the root of the chart under expansion, progressively

**Algorithm 2:** Generation of a new chart.

---

**GenerateNewChart**( $\mathcal{C}_i, \mathbf{F}$ )  
**input** : A chart to expand  $\mathcal{C}_i = \{\mathbf{x}_i, \Phi_i, \mathcal{B}_i, \mathcal{P}_i\}$ , and a set of constraints  $\mathbf{F}$ .  
**output**: A new chart  $\mathcal{C}_j$

```

1  $\mathcal{C}_j \leftarrow \emptyset$ 
2  $\mathbf{u}_i^j \leftarrow \text{RANDOMINBALL}(\mathcal{B}_i)$ 
3 if  $\mathbf{u}_i^j \in \mathcal{P}_i$  then
4    $e \leftarrow \text{TRUE}$  // Small error with respect to  $\mathcal{C}_i$ 
5    $c \leftarrow \text{TRUE}$  // Collision free
6    $t \leftarrow \text{TRUE}$  // Tangent space similar to  $\mathcal{C}_i$ 
7    $s \leftarrow \delta$ 
8   while  $s \leq r$  and  $e$  and  $c$  and  $t$  do
9      $\hat{\mathbf{x}}_j \leftarrow \mathbf{x}_i + \Phi_i s \mathbf{u}_i^j / r$ 
10     $\mathbf{x}_j \leftarrow \text{PROJECT}(\mathcal{C}_i, \hat{\mathbf{x}}_j, \mathbf{F})$ 
11    if  $\|\hat{\mathbf{x}}_j - \mathbf{x}_j\| > \sigma$  then
12       $e \leftarrow \text{FALSE}$ 
13    else
14      if  $\text{COLLISION}(\mathbf{x}_j)$  then
15         $c \leftarrow \text{FALSE}$ 
16      else
17         $\Phi_j \leftarrow \text{TANGENTSPACE}(\mathbf{x}_j, \mathbf{F})$ 
18        if  $\|\Phi_i^\top \Phi_j\| < 1 - \sigma$  then
19           $t \leftarrow \text{FALSE}$ 
20        else
21           $\mathcal{C}_j \leftarrow \text{INITMAP}(\mathbf{x}_j, \mathbf{F})$  //  $\mathcal{C}_j = \{\mathbf{x}_j, \Phi_j, \mathcal{B}_j, \mathcal{P}_j\}$ 
22           $s \leftarrow s + \delta$ 
23 RETURN( $\mathcal{C}_j$ )

```

---

moving to the target point with incremental steps of size  $\delta$ . At each step, we project the point from the tangent space to the manifold (lines 9-10), implementing Eqs. 4 and 5. If the projection converges to a point in the manifold, we check whether the obtained point is not too far away from the tangent space (line 11), whether it is collision free (line 14), and whether the tangent space at the new point (computed using Eq. 2) and that of  $\mathcal{C}_i$  are too different (line 17). In any of these cases, the progress towards the new point is stopped (lines 12, 15, and 19) and we return the chart for the last valid point (line 23), if any.

The main operations of the HC-planner scale as follows. The initialization of a chart scales with  $O(n^3 + 2^k)$ , with  $n$  the dimensionality of the ambient space and  $k$  the dimensionality of the configuration space, since we use a QR decomposition to identify a base of the kernel of the Jacobian of  $\mathbf{F}$  and we have to define a box with  $2^k$  vertices. Both the initialization and the extraction of

an element from a heap are  $O(1)$ . The atlas expansion scales with  $O(n^3)$  since it includes the 5 following steps: 1) the generation of a random number on a  $k$ -dimensional ball ( $O(k)$ ); 2) the check to determine if a point is inside a  $k$ -dimensional polytope (also  $O(k)$  assuming a bounded number of faces per polygon); 3) the projection of a point from the tangent space to the manifold (a Newton process with a bounded number of iterations where at each iteration we use a LU decomposition that is  $O(n^3)$ ); 4) the determinant of a matrix of size  $k$  ( $O(k^3)$ ) that comes from the product of  $n \times k$  matrices ( $O(k^2 n)$ ); and 5) the initialization of a map ( $O(n^3)$ ). When adding a chart to the atlas, we have to look for neighboring charts. This can be done in  $O(k)$  since it is logarithmic with the number of charts that, in the worst case scales exponentially with  $k$ . For the neighboring charts, we have to crop the corresponding polytopes. This operation scales with the number of vertices of those polytope which is  $O(2^k)$ . Finally, the addition of an element in the heap is  $O(k)$ , in agreement with the cost of determining the neighboring relations between charts.

Summarizing, if  $l$  is the number of charts needed to connect the start and the goal the overall algorithm scales with  $O(l(n^3 + 2^k))$ . In the worst case, the final atlas might include all the possible charts for a given manifold. However, as we show in next Section, many problems require in practice a limited number of charts to connect the start and goal configurations.

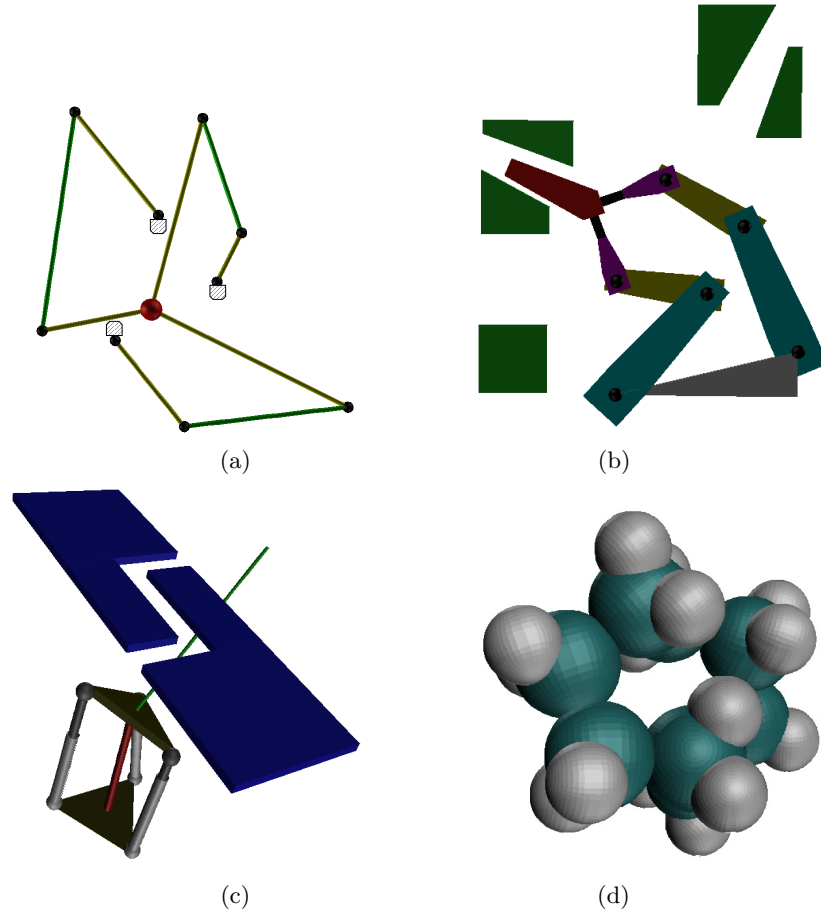
Note that the planner is resolution complete, in the sense that by taking a radius  $r$  small enough for the balls controlling the size of the charts, we can ensure to find a solution path if it exists. In particular, in a problem involving a narrow passage of size  $v$ , charts build from a setting  $r < v/2$  would ensure a solution. However, in practice, much larger radius can be used safely.

## 4 Experiments

We implemented in C the higher dimensional continuation tools<sup>1</sup> described in Section 2 and the HC-planner described in Section 3, both including the treatment of singularities inside the manifold. They were integrated as modules of our position analysis toolbox [30] using SOLID [35, 32] as a collision detector and the GNU Scientific Library for the lineal algebra operations. Our position analysis toolbox is based on a formulation that yields a system of simple equations only containing linear, bilinear, and quadratic monomials, and trivial trigonometric terms for the helical pair only [19]. The simplicity of the final system of equations makes it advantageous for continuation methods [36]. For the purpose of comparison, we also implemented the RRT for constrained spaces presented in [7], which is very similar to the planner in [2]. In this RRT, points are sampled in the ambient space and the nearest sample on the variety is progressively extended towards the random sample. At

---

<sup>1</sup> An implementation of these tools tailored for dynamical systems is available in [31].



**Fig. 5.** The four benchmarks used (a) A star-shaped planar manipulator with three fixed extremities. (b) Two arms collaborating to move an object from one gap to another. (c) A rotational-only parallel manipulator. (d) A cyclooctane molecule.

each extension step the points are projected to the variety using the Jacobian pseudo inverse method. In our implementation, the nearest-neighbor queries use the kd-tree described in [40]. Experiments were executed on a Intel Core 2 at 2.4 Ghz running Linux. Finally, the algorithm parameters were set to  $r = 0.4$ ,  $\delta = 0.04$ ,  $\sigma = 0.1$ , and  $\beta = 1.1$  for all the experiments.

Fig. 5 shows the four benchmarks used in this paper. The first one is a planar star-shaped manipulator also used in [26]. In this case, obstacles are not considered. The second problem involves a system where two arms have to cooperate to move an object from one gap to another. This problem previously appears in [9]. The movement between the start and goal configurations require to traverse singularities, which makes the problem unsolvable

**Table 1.** Dimensionality of the ambient and configuration spaces, execution times and number of samples/charts used by a RRT and the HC-planner.

Benchmarks	$k$	$n$	HC		RRT		HC/RRT	
			Time	Charts	Time	Samples	T/T	C/S
Star-shaped	5	18	1.22	174	1.77	2020	0.69	0.09
Two Arms	3	10	1.26	494	24.04	26697	0.05	0.02
Parallel	3	27	2.45	286	24.38	9454	0.1	0.03
Cyclooctane	2	17	1.13	174	12.1	9814	0.09	0.02

by inverse-kinematic PRM approaches [6, 10]. The third example, kindly provided by Juan Cortés, is a parallel platform with rotation motion only. The task here is to move a stick attached to the robot across some obstacles. The last benchmark is the cyclooctane, a molecule whose kinematics is a 8-revolute chain. Here, we have to find a path between two conformations that avoids self-collisions involving carbon and hydrogen atoms (depicted in cyan and white in the figure, respectively).

Table 1 shows the performance comparison, averaged over 10 runs, between the HC-planner and RRT. For each of the four benchmarks, the table gives the dimensionality of the configuration space ( $k$ ), the dimensionality of the ambient space ( $n$ ), the execution times and the number of charts or samples used for each method. The table also shows execution time ratios (T/T) and the ratio between the number charts used in the HC-planner and the number of sampled used with RRT (C/S). Note that the RRT in the Two Arms test case is unable to find a solution with less than 50000 samples for 30% of the cases. In Table 1, the RRT results for this problem correspond to averages for the successful tests only.

The results show how the larger the difference between the ambient and the configuration spaces the higher the advantage of using the HC-planner. Actually, the execution time of the HC-planner is an order of magnitude smaller than that of the RRT except for the Star-shaped problem where the HC-planner is only about a 40% faster. This tendency is true despite the generation of samples being much faster than the generation of charts. This is so because charts are more powerful since they do not only describe the variety on a single point but on a local neighborhood of a point. Thus, the HC-algorithm uses in average only about a 4% of the number of samples used by RRT.

## 5 Conclusions

In this paper, we extended the use of High-dimensional Continuation algorithmic tools for path planning applications. Using these tools, we defined a randomized path planner for highly constrained systems. The presented plan-

ner directly works on the configuration space, trying to connect any pair of query configurations with a small collection of local charts. The algorithm performance is highly independent of the relation between the configuration space and the ambient space. This is in contrast with existing sampling algorithms for constrained problems that generate samples in the ambient space. The experiments show that our approach can be up to one order of magnitude faster than existing algorithms.

The cost of the algorithm introduced in this paper is exponential with the dimension of the configuration space, which is in agreement with the cost of the best complete path planners [4]. Thus, the algorithm would not scale gracefully to high-dimensional problems. Despite this, the use of a greedy search strategy together with the randomization allow to solve problems with moderate complexity (at least up to dimension 5 in the examples) embedded in even higher-dimensional spaces. Problems slightly more complex than this are also likely to be addressable with the presented planner and this includes many interesting problems in Robotics and in Molecular Biology [3]. To scale to problems with even larger dimensionality we could rely on charts with larger area of influence. However this is likely to be valid only in almost lineal problems, where the error between the tangent space and the solution variety remains small over large areas. Another possibility is to define a cost function over the configuration space so that the exploration could be limited to areas with low cost [16]. All these points deserve a more careful evaluation. It is also our future endeavor to perform a more thorough experimental and theoretical analysis of the proposed algorithm.

## Acknowledgments

We would like to thank L. Ros for pointing us to the Higher-dimension Continuation tools and for fruitful discussions during the elaboration of this work. This work has been partially supported by the Spanish Ministry of Science and Innovation under project DPI2007-60858.

## References

1. G. Ballantyne and F. Moll. The da Vinci telerobotic surgical system: Virtual operative field and telepresence surgery. *Surgical Clinics of North America*, 83(6):1293–1304, 2003.
2. D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner. Manipulation planning on constraint manifolds. In *IEEE International Conference on Robotics and Automation*, pages 1383–1390, 2009.
3. W. M. Brown, S. Martin, S. N. Pollock, E. A. Coutsiias, and J.-P. Watson. Algorithmic dimensionality reduction for molecular structure analysis. *Journal of Chemical Physics*, 129(6):064118, 2008.
4. J. Canny. *The Complexity of Robot Motion Planing*. MIT Press, 1988.

5. P.-C. Chen, P. Hansen, and B. Jaumard. On-line and off-line vertex enumeration by adjacency lists. *Operation Research Letters*, 10:403–409, 1991.
6. J. Cortés, T. Siméon, and J. P. Laumond. A random loop generator for planning the motions of closed kinematic chains using PRM methods. In *IEEE International Conference on Robotics and Automation*, pages 2141–2146, 2002.
7. S. Dalibard, A. Nakhaei, F. Lamiroux, and J.-P. Laumond. Whole-body task planning for a humanoid robot: a way to integrate collision avoidance. In *IEEE-RAS International Conference on Humanoid Robots*, pages 355–360, 2009.
8. G. F. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer, 1996.
9. M. Gharbi, J. Cortés, and T. Siméon. A sampling-based path planner for dual-arm manipulation. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 383–388, 2008.
10. L. Han and N. M. Amato. A kinematics-based probabilistic roadmap method for closed chain systems. In *Algorithmic and Computational Robotics - New Directions (WAFR2000)*, pages 233–246, 2000.
11. L. Han and L. Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics: Science and Systems*, pages 177–184, 2005.
12. I. Havoutis and S. Ramamoorthy. Motion synthesis through randomized exploration of submanifolds of configuration spaces. In *RoboCup 2009: Robot Soccer World Cup XIII. Lecture Notes in Artificial Intelligence*, volume 5949, pages 92–103, 2009.
13. M. E. Henderson. Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos*, 12(3):451–476, 2002.
14. M. E. Henderson. Multiparameter parallel search branch switching. *International Journal of Bifurcation and Chaos in Applied Science and Engineering*, 15(3):967–974, 2005.
15. M. E. Henderson. *Numerical continuation methods for dynamical systems: path following and boundary value problems*, chapter Higher-Dimensional Continuation. Springer, 2007.
16. L. Jaillet, J. Cortés, and T. Siméon. Sampling-based path planning on configuration-space costmaps. *IEEE Transactions on Robotics*, 2010.
17. B. Krauskopf, H. M. Osinga, and J. Galán-Vioque. *Numerical continuation methods for dynamical systems: path following and boundary value problems*. Springer, 2007.
18. C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schafer, B. Brunner, H. Hirschmuller, and G. Hirzinger. A humanoid two-arm system for dexterous manipulation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 276–283, 2006.
19. J. M. Porta, L. Ros, and F. Thomas. A linear relaxation technique for the position analysis of multiloop linkages. *IEEE Transactions on Robotics*, 25(2):225–239, 2009.
20. A. Pressley. *Elementary Differential Geometry*. Springer Verlag, 2001.
21. W. C. Rheinboldt. MANPACK: A set of algorithms of computations on implicitly defined manifolds. *Computers and Mathematics with Applications*, 32(12):15–28, 1996.
22. A. Rodríguez, L. Basañez, and E. Celaya. A relational positioning methodology for robot task specification and execution. *IEEE Transactions on Robotics*, 24(3):600–611, 2008.

23. C. Rosales, L. Ros, J. M. Porta, and R. Suárez. Synthesizing grasp configurations with specified contact regions. *International Journal of Robotics Research*, 2010.
24. B. Roth and F. Freudenstein. Synthesis of path-generating mechanisms by numerical methods. *ASME Journal of Engineering for Industry*, 85:298–307, 1963.
25. A. Shkolnik and R. Tedrake. Path planning in 1000+ dimensions using a task-space voronoi bias. In *IEEE International Conference on Robotics and Automation*, pages 2892–2898, 2009.
26. N. Shvlab, G. Liu, M. Shoham, and J. C. Trinkle. Motion planning for a class of planar closed-chain manipulators. *International Journal of Robotics Research*, 26(5):457–473, 2007.
27. A. J. Sommese and C. W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005.
28. M. Stilman. Task constrained motion planning in robot joint space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3074–3081, 2007.
29. X. Tang, S. Thomas, P. Coleman, and N. M. Amato. Reachable distance space: Efficient sampling-based planning for spatially constrained systems. *International Journal of Robotics Research*, 29(7):916–934, 2010.
30. The CUIK project web page. <http://www.iri.upc.edu/research/webprojects/cuikweb>.
31. The MultiFario project web page. <http://multifario.sourceforge.net>.
32. The SOLID web page. <http://www.dtecta.com>.
33. L.-W. Tsai. *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley and Sons, 1999.
34. T. T. Um, B. Kim, C. Suh, and F. C. Park. Tangent space RRT with lazy projection: An efficient planning algorithm for constrained motions. In *Advances in Robot Kinematics*, pages 251–260, 2010.
35. G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.
36. C. Wampler and A. Morgan. Solving the 6R inverse position problem using a generic-case solution methodology. *Mechanism and Machine Theory*, 26(1):91–106, 1991.
37. W. J. Wedemeyer and H. Scheraga. Exact analytical loop closure in proteins using polynomial equations. *Journal of Computational Chemistry*, 20(8):819–844, 1999.
38. J. H. Yakey, S. M. LaValle, and L. E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE Transactions on Robotics and Automation*, 17(6):951–959, 2001.
39. Z. Yao and K. Gupta. Path planning with general end-effector constraints: Using task space to guide configuration space search. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1875–1880, 2005.
40. A. Yershova and S. M. LaValle. Improving motion planning algorithms by efficient nearest neighbor searching. *IEEE Transactions on Robotics*, 23(1):151–157, 2007.
41. A. Yershova and S. M. LaValle. Motion planning for highly constrained spaces. In *Robot Motion and Control. Lecture Notes on Control and Information Sciences*, volume 396, pages 297–306, 2009.