

# Randomized Path Planning on Manifolds based on Higher-Dimensional Continuation

Josep M. Porta, Léonard Jaillet, and Oriol Bohigas

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

## Abstract

*Despite the significant advances in path planning methods, highly constrained problems are still challenging. In some situations, the presence of constraints defines a configuration space that is a non-parametrizable manifold embedded in a high dimensional ambient space. In these cases, the use of sampling-based path planners is cumbersome since samples in the ambient space have low probability to lay on the configuration space manifold. In this paper, we present a new path planning algorithm specially tailored for highly constrained systems. The proposed planner builds on recently developed tools for higher-dimensional continuation, which provide numerical procedures to describe an implicitly defined manifold using a set of local charts. We propose to extend these methods focussing the generation of charts on the path between the two configurations to connect and randomizing the process to find alternative paths in the presence of obstacles. The advantage of this planner comes from the fact that it directly operates into the configuration space and not into the higher-dimensional ambient space, as most of the existing methods do.*

## Keywords

Path Planning, Constraints, Manifolds, Higher-Dimensional Continuation.

## 1 Introduction

Many problems require to connect two points with a path fulfilling some constraints expressed as a set of equations. In Robotics, this appears, for instance, in parallel manipulators (Tsai, 1999), robot grasping (Rosales et al., 2011), constraint-based object positioning (Rodríguez et al., 2008), surgery robots (Baltayne and Moll, 2003), and humanoid robots (Ott et al., 2006). This situation also appears in Biochemistry when searching for low energy paths between different molecular conformations (Wedemeyer and Scheraga, 1999). In all these cases, the set of points fulfilling the constraints define a configuration space that is a variety composed by one or more manifolds embedded in the higher-dimensional ambient space of the variables involved in the equations (Lavalle, 2011).

The adaptation of otherwise successful sampling-based path planning methods (Kavraki et al., 1996; LaValle, 2006) is not straightforward since samples in the ambient space have a null probability to lay on the configuration space. Consequently, several methods have been devised to find points on the configuration space from points of the ambient space. All these approaches, however, only perform properly when the ambient and the configuration spaces are similar. If the constraints define a complex surface 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 these approaches. This problem may appear even in simple cases such as the one described in Fig. 1, where a Rapidly-exploring Random Tree (RRT) is built on a sphere using a recent method for constrained probabilistic path planning (Dalibard et al., 2009). This method selects points in the ambient space (a box in  $\mathbb{R}^3$  including the sphere in this case) and uses iterative numerical methods to converge to the configuration space (the sphere in the example). If the sphere is not centered in, and tightly enveloped by the ambient space box, the sampling process is biased and the result is a poor exploration of the configuration space. The lack of prior knowledge about the configuration space structure and its relation to the ambient space makes it hard to forecast whether or not a sampling-based planning approach would be adequate.

Ideally, one would like to sample directly on the configuration space and not in the ambient space. Unfortunately, a uniform sampling over this space typically relies on a global parametrization that is generally not available. However, from Differential Geometry, it is well known that a manifold can be described by an atlas containing a collection of charts, each chart providing a local parametrization (Pressley, 2001). Higher-

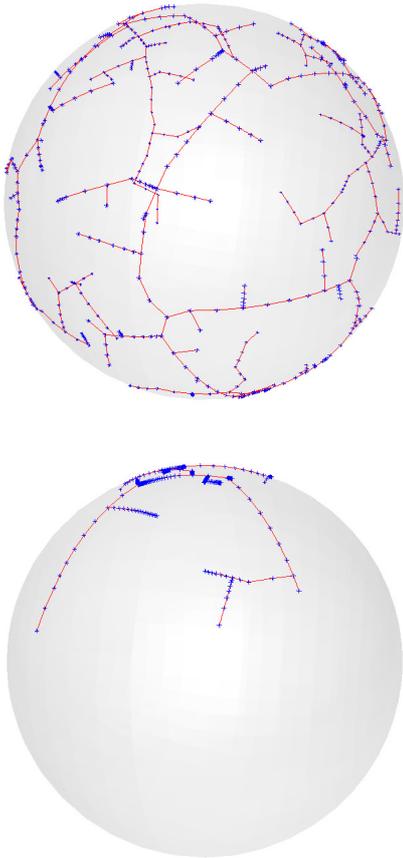


Figure 1: Exploration trees with 500 samples. Blue crosses represent the tree nodes and red lines the connections between them. **Top** When the ambient space is a box tightly enveloping the sphere, the exploration is relatively homogeneous. **Bottom** When the box is elongated, for instance along the positive vertical axis, the exploration is hindered.

dimensional continuation techniques provide principled numerical tools to compute the atlas of one of the connected components of an implicitly defined manifold, departing from a point. For instance, Fig. 2-top shows the atlas obtained with the most recent of these techniques (Henderson, 2002a) in the toy problem of the sphere.

In this paper, we extend the tools developed for higher-dimensional continuation to the context of path planning for highly constrained systems. We define the concept of partial atlas connecting two configurations, dealing with the presence of obstacles. We also introduce the random exploration of a manifold focused towards a target configuration. As a result, we obtain a higher-dimensional continuation planner (HC-planner) that clearly outperforms existing approaches. Figure 2-bottom shows an example of path found with

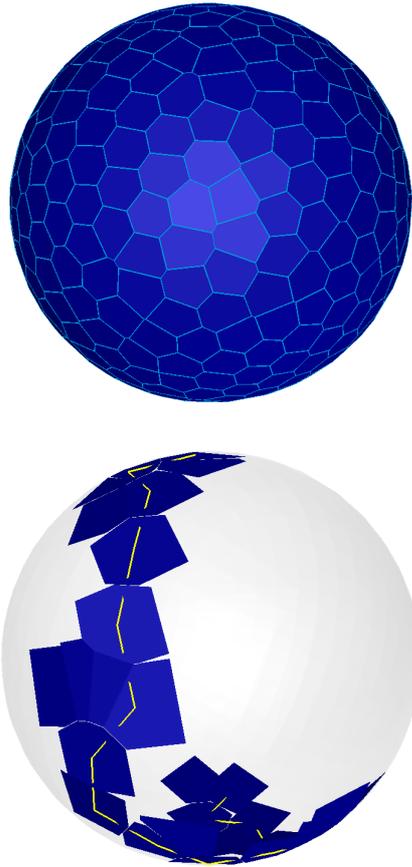


Figure 2: Atlas of the sphere obtained by higher-dimensional continuation. Each polygon represents a chart that locally parametrizes the sphere. **Top** The full atlas includes about 500 charts. **Bottom** The partial atlas built with our planning approach when connecting the two poles. Only about 30 charts are generated. The solution path is shown as a yellow line.

our approach for the sphere toy problem when trying to connect the two poles. Note how only a small set of charts is needed to find a path connecting the query points.

Next section provides a review of path planning techniques for constrained problems. Then, we introduce the higher-dimensional continuation tools that are extended in Section 4 to the context of path planning. Section 5 compares the performance of the planner with respect to existing methods for several benchmarks. Finally, Section 6 gives an overview of the contributions of this work and indicates points that deserve further attention.

## 2 Related Work

Some approaches try to explicitly describe the configuration space manifolds arising in constrained path planning problems, but they are either too complex to be applied in practice (Canny, 1988), or limited to particular architectures (Shvlab et al., 2007). Thus, the usual approach to address these problems is to extend the common sampling-based planning methods. The performance of those methods heavily rely on being able to uniformly sample on the space to explore. In some families of mechanism, distance-based formulations provide a global parametrization of the configuration space manifold that can be straightforwardly used to uniformly sample this space (Han et al., 2008; Tang et al., 2010). However, a global isometric parametrization of the configuration space is not available in general and alternative sampling strategies have been devised.

In this direction, the Kinematics-based Roadmap method (Han and Amato, 2000) samples a subset of variables corresponding to the so called active part of the mechanism and uses inverse kinematics to find all the possible values for the remaining ones, i.e., those for the passive part of the mechanism. However, this strategy is only valid for particular families of mechanisms. Moreover, although some improvements have been proposed (Cortés et al., 2002), the probability of generating invalid samples is significant. Finally, the several solutions of the inverse kinematic functions and their singularities complicates the approach (Gharbi et al., 2008). Similar issues arise in task-space planning (Yao and Gupta, 2005; Shkolmik and Tedrake, 2009), where sampling is performed in a subset of variables associated with the end effector of the robot and the rest of variables are iteratively determined.

An alternative strategy to get a valid configuration is to sample in the ambient space and use numerical techniques to project the samples to the configuration space. In this way, the need to rely on inverse kinematic functions is avoided. Yakey et al. (2001) propose the Randomized Gradient Descent (RGD) method where the sample is randomly perturbed, keeping only the perturbations that reduce the error with respect to the configuration space. Stilman (2007, 2010) proposes two projections strategies. The first one is the Tangent Space sampling that projects the sample to the tangent space of the nearest node already on the configuration space and, from this projection, executes RGD until the configuration space is reached. The second strategy is the First-order Retraction and uses the Jacobian pseudo-inverse method (Whitney, 1969) to converge to the manifold. Stilman (2007, 2010) uses a RRT-extend strategy and suggests the possible use of a RRT-connect procedure. This latter procedure has been recently implemented by Dalibard et al. (2009)

and by Berenson et al. (2009, 2011) in two different ways and in the context of planning with torque limits and task-constrained planning, respectively. Moreover, Berenson et al. (2009, 2011) use a bidirectional RRT instead of a single RRT, yielding significant performance improvements. Although these methods can guarantee probabilistic completeness (Berenson et al., 2011), their performance depends on the similarity between the ambient and the configuration spaces. If the configuration space is heavily folded, as it is the case in many realistic situations, samples do not uniformly distribute on the configuration space, hindering its efficient exploration using these methods.

One way to alleviate the problems of mismatching between the two spaces is to learn a model of the configuration space (Havoutis and Ramamoorthy, 2009). The learned model can then be exploited to compute geodesic distances between configurations, and to make projections and expansions that better fit the configuration space shape. However, the learning process can only deal with relatively simple manifolds and it typically requires a large set of given samples on the manifold, which makes this approach computationally very expensive.

Another possibility is to focus the sampling on a subset of the ambient space around the configuration space (Yershova and LaValle, 2009). However, even in the case where the configuration is properly bounded, samples are thrown in the ambient space that can be of much higher dimensionality than the configuration space. Um et al. (2010) sketch an appealing lazy RRT scheme where the tree is defined in the tangent of the configuration space, which is of the same dimensionality as the manifold. However, the approach uses points on the tangent space that do not actually fulfill the equations defining the configuration space, it does not consider the presence of singularities in the manifold, and it introduces an overlap between tangent spaces that can lead to an inappropriate sampling bias.

Although in general a global parametrization of the configuration space is not available, local parametrizations that can be coordinated to fully describe a manifold are theoretically introduced in Differential Geometry and practically implemented in the so called continuation methods, which have been strongly developed in the context of Dynamical Systems (Krauskopf et al., 2007; Henderson, 2007). In Robotics, one-dimensional continuation methods (also known as path following, homotopy or bootstrap methods) have been mainly used for solving problems related to Kinematics (Roth and Freudenstein, 1963; Sommese and Wampler, 2005). To the best of our knowledge, higher-dimensional continuation methods have been only used marginally in Robotics to evaluate the dexterity of mechanisms (Yang and Haug, 1994) and in a preliminary version of the planner described here (Porta and

Jaillet, 2010). The work by Yang and Haug (1994), though, uses the Moving Frame algorithm (Rheinboldt, 1988) that defines a collection of simplexes in the ambient space that fully bound the connected component of the configuration space including a given initial point, an approach similar to that by Yershova and LaValle (2009). The advantage of the more recent higher-dimensional continuation techniques that we use herein (Henderson, 2002a) is that they directly operate on the tangent space associated to the configuration space, which has the same dimensionality as the configuration space and that, in the problems we focus on, has much lower dimensionality than the ambient space.

### 3 Higher-dimensional Continuation

Next, we describe the main algorithmic tools introduced by Henderson (2002a) which generalize the one-dimensional pseudo-arclength procedure (Keller, 1977). These tools generate an atlas for one of the connected components of a  $k$ -dimensional continuously differentiable variety implicitly defined in a  $n$ -dimensional space by a system of equations

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

with  $\mathbf{F} : \mathbb{R}^n \rightarrow \mathbb{R}^{n-k}$ ,  $n > k > 0$ ,  $n$  the dimension of the ambient space and  $k$  the dimension of the variety. To simplify the description, we will further assume that the variety implicitly defined by  $\mathbf{F}$  is actually a manifold until Section 3.3, where we consider singularities leading to bifurcations. Figure 3 illustrates the main idea on which relies the approach.

#### 3.1 Defining a Chart

Given a point on the manifold,  $\mathbf{x}_i$ , a chart,  $\mathcal{C}_i$ , defines a mapping  $\psi_i : \mathbb{R}^k \rightarrow \mathbb{R}^n$  from parameters in  $\mathbb{R}^k$  to a portion of the manifold around  $\mathbf{x}_i$ , with  $\psi_i(\mathbf{0}) = \mathbf{x}_i$ . Henderson (2002a) defines the mapping  $\psi_i$  using  $\Phi_i$ , an orthonormal basis of the tangent space of the manifold at  $\mathbf{x}_i$ . This basis 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  $k \times k$  identity matrix. Then, the mapping  $\mathbf{x}_j = \psi_i(\mathbf{u}_i^j)$  is implemented by first selecting a  $k$ -dimensional vector  $\mathbf{u}_i^j$  of parameters in tangent space, that is used to generate a point  $\hat{\mathbf{x}}_i^j$  in ambient space as

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

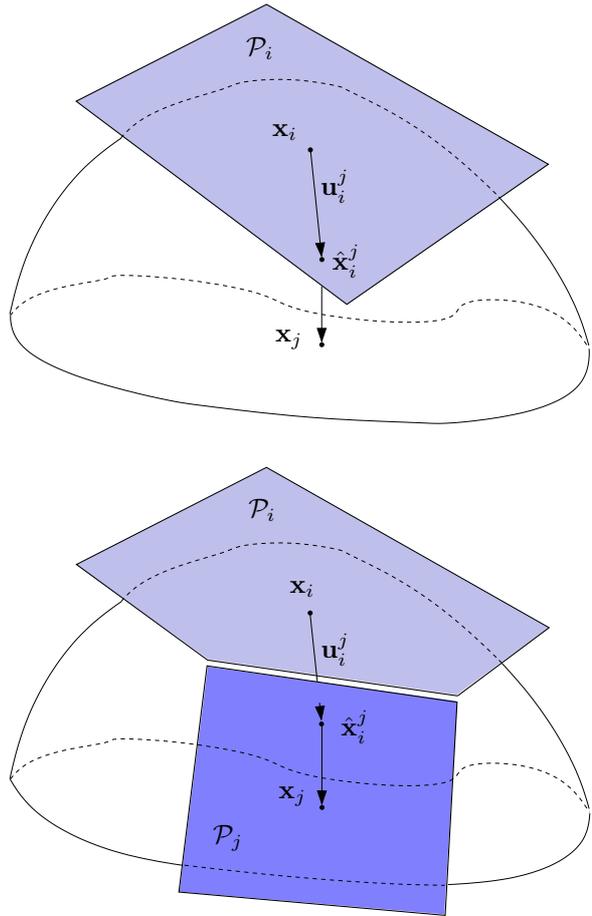


Figure 3: Higher-dimensional continuation method applied to a two-dimensional manifold embedded in a 3D ambient space. **Top** A chart at a given point,  $\mathbf{x}_i$ , is defined using the tangent space at this point. The area of applicability of the chart is denoted as  $\mathcal{P}_i$ . A point,  $\hat{\mathbf{x}}_i^j$ , is defined using the tangent space and is orthogonally projected to the manifold to determine  $\mathbf{x}_j$ , the center of a neighboring chart. **Bottom** The area of applicability of the new chart,  $\mathcal{P}_j$ , is coordinated with  $\mathcal{P}_i$  so that their projections to the manifold slightly overlap.

Then,  $\mathbf{x}_j$  is the orthogonal projection of  $\hat{\mathbf{x}}_i^j$  on the manifold. This projection is obtained by solving the system (Rheinboldt, 1996)

$$\left. \begin{aligned} \mathbf{F}(\mathbf{x}_j) &= \mathbf{0} \\ \Phi_i^\top (\mathbf{x}_j - \hat{\mathbf{x}}_i^j) &= \mathbf{0} \end{aligned} \right\} \quad (4)$$

using a Newton procedure where  $\mathbf{x}_j$  is initialized to  $\hat{\mathbf{x}}_i^j$  and iteratively updated with the increment  $\Delta \mathbf{x}_j$  fulfilling

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

The update is applied until the norm of the right-hand side of the previous system becomes negligible or for a

maximum number of iterations. If the process is not convergent, the input parameter  $\mathbf{u}_i^j$  is out of  $\mathcal{P}_i$ , the applicability area for chart  $\mathcal{C}_i$ . The areas out of the scope of the chart are to be parametrized by other charts.

The inverse of mapping of  $\psi_i$  can be computed as

$$\mathbf{u}_i^j = \psi_i^{-1}(\mathbf{x}_j) = \Phi_i^\top(\mathbf{x}_j - \mathbf{x}_i), \quad (6)$$

and can be applied for any point  $\mathbf{x}_j$  in  $\mathbb{R}^n$ , regardless of whether or not it is on the manifold or if it is actually projected into  $\mathcal{P}_i$ .

### 3.2 Defining an Atlas

Since the applicability area for each chart is limited, the full parametrization of the manifold requires to define an atlas, i.e., a collection of charts properly coordinated. The algorithm proposed by Henderson (2002a) gives a systematic way to add new charts to the atlas and to bound their associated applicability areas. In this work,  $\mathcal{P}_i$  is represented as a polytope which is initialized as an hypercube enclosing a ball,  $\mathcal{B}_i$ , of radius  $r$ .  $\mathcal{P}_i$  is progressively refined as new charts are added to the atlas, as illustrated in Fig. 4. A vector,  $\mathbf{v}$ , to the first vertex of  $\mathcal{P}_i$  external to  $\mathcal{B}_i$  is used to generate a new chart. From  $\mathbf{v}$ , a vector of local parameters giving a point on  $\mathcal{B}_i$  is computed as

$$\mathbf{u}_i^j = \alpha \frac{r}{\|\mathbf{v}\|} \mathbf{v}, \quad (7)$$

with  $\alpha$  initialized to 1. A new neighboring chart,  $\mathcal{C}_j$ , is then defined on the point  $\mathbf{x}_j$  resulting from applying the mapping  $\psi_i(\mathbf{u}_i^j)$ . If  $\mathcal{C}_i$  and the new  $\mathcal{C}_j$  are too far or too different, i.e., if

$$\|\mathbf{x}_j - \hat{\mathbf{x}}_i^j\| > \sigma, \quad (8)$$

or if

$$\|\Phi_i^\top \Phi_j\| < 1 - \sigma, \quad (9)$$

for a given parameter  $\sigma$ , 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  $\mathbf{x}_i$ , i.e., with a smaller  $\lambda$  in Eq. (7). This adapts the distribution and the size of the applicability areas of the charts to the local curvature of the manifold. Each new chart,  $\mathcal{C}_j$ , added to the atlas has to be properly coordinated with the applicability areas of those charts already in the atlas. In the example in Fig. 4,  $\mathcal{C}_j$  is used to refine  $\mathcal{P}_i$  from the intersection between  $\mathcal{B}_i$  and  $\tilde{\mathcal{B}}_i^j$ , a ball of radius  $r$  that approximates  $\hat{\mathcal{C}}_i^j$ , the projection into the tangent space of  $\mathcal{C}_i$  of the part of the manifold covered by  $\mathcal{C}_j$ , as shown in Fig. 4-bottom.

The hyperplane defined by the intersection of  $\mathcal{B}_i$  and  $\tilde{\mathcal{B}}_i^j$  can be computed by subtracting the equations for the two balls. As shown in Fig. 4-bottom, this plane defines a new face of  $\mathcal{P}_i$  that eliminates some of its vertices (in particular the one giving  $\mathbf{v}$ ) and generates new

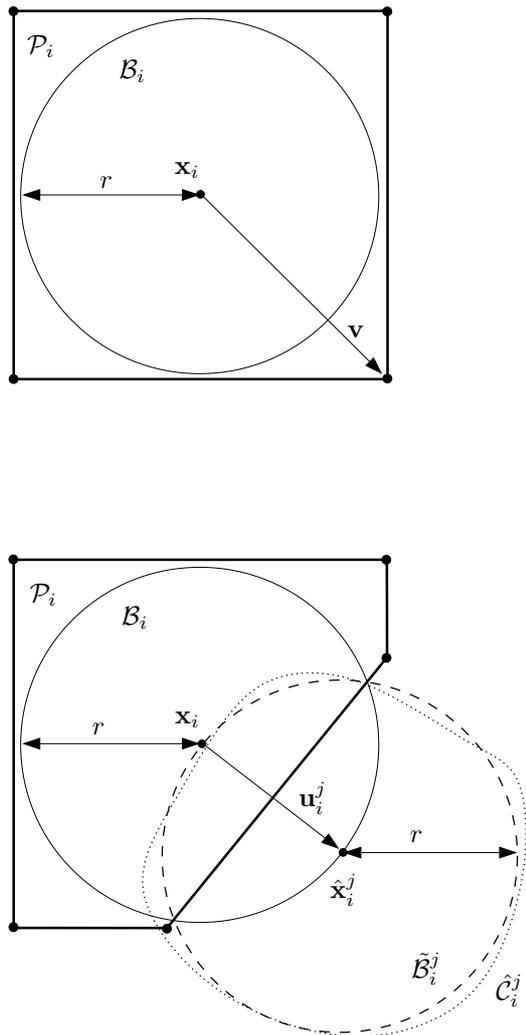


Figure 4: Progressive refinement of the applicability area of a chart of a 2D manifold. **Top** The applicability area,  $\mathcal{P}_i$ , of chart  $\mathcal{C}_i$  is initialized as a box including a ball of radius  $r$  around  $\mathbf{x}_i$ . **Bottom** The polytope is refined using a ball  $\tilde{\mathcal{B}}_i^j$  that approximates  $\hat{\mathcal{C}}_i^j$ , the projection of the area of the manifold covered by a neighboring chart into the current chart.

ones. Similarly,  $\mathcal{P}_j$ , the polytope associated to  $\mathcal{B}_j$ , is cropped using an approximation of the projection of  $\mathcal{C}_i$  into  $\mathcal{C}_j$ . The applicability areas of the two neighboring charts are not necessarily continuous, but under mild conditions (Henderson, 2002a) their projection to the manifold slightly overlaps, smoothly covering it. The transition from one local parametrization to the other can be done using the direct and inverse mappings for the two neighboring charts. For a given vector of parameters  $\mathbf{u}_i$  at the border of  $\mathcal{P}_i$ , the corresponding vec-

tor of parameters in chart  $\mathcal{C}_j$  is

$$\mathbf{u}_j = \psi_j^{-1}(\psi_i(\mathbf{u}_i)). \quad (10)$$

When a chart is fully surrounded by other charts, all the vertexes of its polytope are inside the associated ball and the chart is not further expanded as its applicability area is considered bounded. This process of chart expansion continues as far as there are non-bounded charts in the atlas. At the end of the process, the connected component of the configuration space containing the initial point is fully covered by a set of charts whose area of validity are bounded by the corresponding polytopes (see Fig. 2-top).

The cost of the algorithm at each step is dominated by the cost of two searches among the set of charts: one to find a non-bounded chart and another to find the potential neighbors of a new chart. The first search can be saved keeping the non-bounded charts in a list. The cost of the second search can be reduced using a kd-tree storing the center points of the charts. We can avoid the definition of the kd-tree in the high-dimensional ambient space where it would be less effective and define it in a  $k$ -dimensional space by projecting the chart centers into the tangent space of a given chart, for instance the first one using  $\psi_1^{-1}$ .

### 3.3 Dealing with Bifurcations

If the configuration space is not a manifold everywhere, we need to consider the presence of singularities. A singularity occurs at points where the Jacobian of the system of equations implicitly defining the configuration space is not full rank. Although many types of singularities exist (Beyn et al., 2002), we only consider situations where the manifold bifurcates in two branches (see Fig. 5-top). In those cases, the continuation process evolving in one of the branches needs to detect the presence of the other branch and to generate a chart on it from where to extend the atlas. Otherwise, the connected component of the configuration space containing the given initial point might not be completely characterized.

The bifurcations define a zero-measure set of dimension  $k - 1$  made of the points where the rank of the Jacobian of  $\mathbf{F}$  is  $n - k - 1$ . Points on this set can be located by monitoring an indicator function,  $\chi(\mathbf{x})$ , whose value is different for two points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  at opposite sides of the bifurcation, and that vanishes at the bifurcation. One possible indicator function is

$$\chi(\mathbf{x}) = \text{sign} \left| \begin{array}{c} \mathbf{J}(\mathbf{x}) \\ \hat{\Phi}_b \end{array} \right|, \quad (11)$$

where  $\hat{\Phi}_b$  is an approximation of the tangent space at the bifurcation for the branch including both  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . This approximation can be computed, for instance, interpolating between  $\Phi_i$  and  $\Phi_j$ . Thus, if, when adding

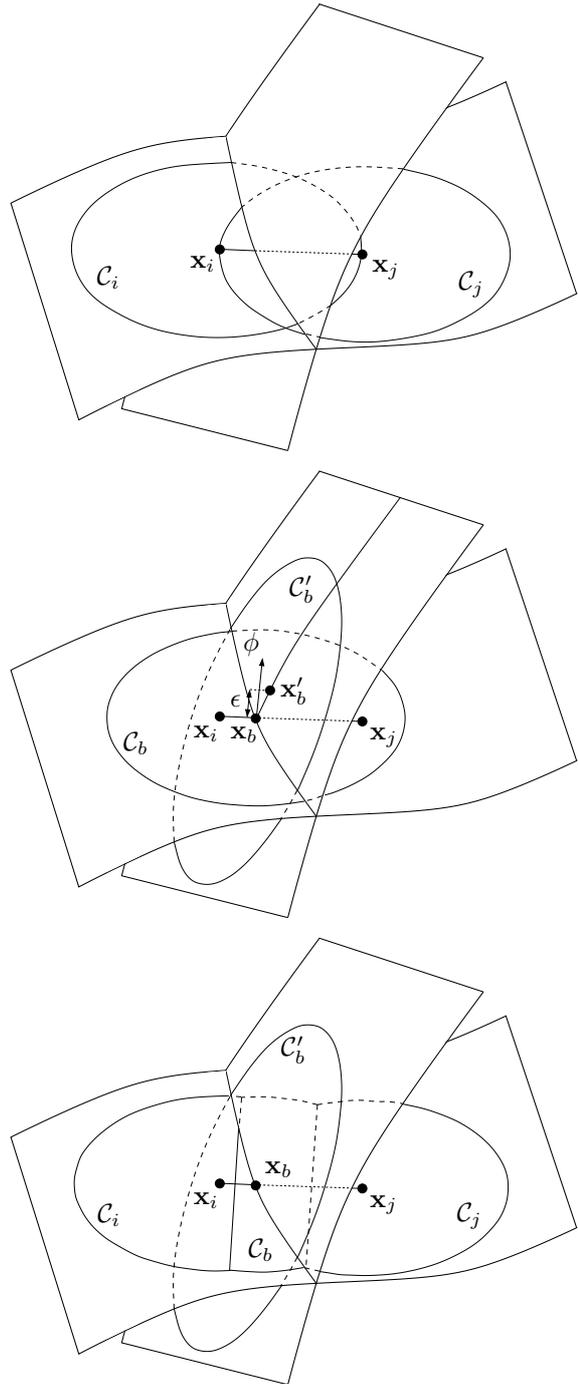


Figure 5: A bifurcation in the configuration space. **Top** The bifurcation is detected between charts  $\mathcal{C}_i$  and  $\mathcal{C}_j$  by monitoring an indicator function  $\chi$ . **Middle** Once a bifurcation point,  $\mathbf{x}_b$ , is located, two charts are defined, one on the current branch,  $\mathcal{C}_b$ , and another on the new branch,  $\mathcal{C}'_b$ . **Bottom** These new charts are intersected with those already in the atlas. In this example,  $\mathcal{C}_b$  is intersected with  $\mathcal{C}_i$  and  $\mathcal{C}_j$ , whereas  $\mathcal{C}'_b$  is the first chart on the new branch.

a chart to the atlas,  $\chi$  varies, the transition between the new chart,  $\mathcal{C}_j$ , and its parent,  $\mathcal{C}_i$ , crosses a bifurcation. In this case and since chart  $\mathcal{C}_j$  is generated from chart  $\mathcal{C}_i$  using  $\mathbf{x}_j = \psi_i(\mathbf{u}_i^j)$ , a point on the bifurcation,  $\mathbf{x}_b$ , is located using a dichotomic search along vector  $\mathbf{u}_i^j$ , defined in the tangent space associated to chart  $\mathcal{C}_i$ . At a bifurcation point, the kernel of the Jacobian of  $\mathbf{F}$  includes  $k+1$  vectors. The first  $k$  vectors,  $\phi_1, \dots, \phi_k$ , correspond to those in  $\hat{\Phi}_b$ . The last vector of the kernel,  $\phi_{k+1}$ , provides a direction to expand the new branch. Moreover, at the singular point, there is a left null vector for the Jacobian,  $\varphi$ , such that  $\varphi^\top \mathbf{J}(\mathbf{x}_b) = 0$  and  $\varphi^\top \varphi = 1$ .

Once the bifurcation point,  $\mathbf{x}_b$ , is accurately determined, two tangents to the configuration space can be defined on it, one for each branch (see Fig. 5-middle). The objective here is to determine those two tangent spaces and use them to define two charts,  $\mathcal{C}_b$  and  $\mathcal{C}'_b$ , at the bifurcation point. As said, the tangent space for the branch containing  $\mathcal{C}_i$  and  $\mathcal{C}_j$  is approximated by  $\hat{\Phi}_b$ . According to Henderson (2005) and Bohigas (2011), if the Hessian of  $\mathbf{F}$  is available, the first  $k-1$  vectors,  $\phi'_1 \dots \phi'_{k-1}$ , of the basis of the tangent space for the new branch,  $\hat{\Phi}'_b$ , are those defining a basis for the subspace of  $\hat{\Phi}_b$  orthogonal to

$$\sum_{i=1}^{k+1} N_i \phi_i, \quad (12)$$

with

$$N_i = \sum_{l=1}^{n-k} \varphi_l \phi_i^\top \mathbf{H}_l(\mathbf{x}_b) \phi_{k+1} \quad (13)$$

and where  $\varphi_l$  is the  $l$ -th component of  $\varphi$  and  $\mathbf{H}_l(\mathbf{x}_b)$  is the Hessian of  $f_l$  (the  $l$ -th function in  $\mathbf{F}$ ) evaluated at  $\mathbf{x}_b$ . The last (not normalized) vector of  $\hat{\Phi}'_b$  is

$$\phi'_k = \sum_{i=1}^k N_{k+1} N_i \phi_i - \left[ \sum_{i=1}^k N_i N_i \right] \phi_{k+1}. \quad (14)$$

If the Hessian is not available,  $\hat{\Phi}'_b$  can be approximated using the tangent space at a point  $\mathbf{x}'_b$  on the other branch, close to  $\mathbf{x}_b$ . This point can be computed solving

$$\left. \begin{aligned} \mathbf{F}(\mathbf{x}) &= \mathbf{0} \\ \phi_{k+1}^\top (\mathbf{x} - \mathbf{x}_b) &= \epsilon \end{aligned} \right\} \quad (15)$$

with  $\epsilon$  small (we use  $10^{-3}$  in our implementation) and using a Newton process initialized at  $\mathbf{x}_b + \epsilon \phi_{k+1}$ , as illustrated in Fig. 5-middle. Vector  $\phi'_k$  can be used instead of  $\phi_{k+1}$  in Eq. (15) to get a closer initial approximation to the new branch. Note that this requires the Hessian, but only to compute  $\phi'_k$  since the rest of  $\phi'$  vectors are not used.

The two charts defined at  $\mathbf{x}_b$  using  $\hat{\Phi}_b$  and  $\hat{\Phi}'_b$  are added to the atlas and intersected with previously

added charts, as shown in Fig. 5-bottom. Moreover, since both charts share the same center, they are considered neighbors and this relation is used to cross the bifurcation when necessary.

## 4 Path-Planning on Manifolds

Using the tools described in the previous section, a graph can be built where nodes are the chart centers and edges represent the neighboring relations between charts. Then, the shortest path connecting two given points can be computed using a standard graph search method such as A\* considering only the collision-free transitions between the chart centers. Note that when applying A\* the atlas is not pre-computed, but incrementally generated as the search progresses. However, in practice, a large portion of the configuration space is explored due to the optimality of A\*. Thus, this method defines an optimal, resolution complete path planner, but it is only practical for low dimensional manifolds, specially if the charts have small applicability areas. If we define charts with coarser resolution, the presence of obstacles becomes an issue since, in an environment with many obstacles, most of the transitions between chart centers will be in collision and it will not be possible to find ways out among obstacles.

Herein, we propose modifications to the higher-order continuation procedures to deal with the curse of dimensionality and the presence of obstacles. First, we capitalize the fact that path planning is only concerned with the path between two given configurations and not with the full atlas, 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 and adapt the generation of charts to the environment structure.

### 4.1 Chart selection: focusing on 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 strategy where the chart to expand is the one with minimum expected cost to reach the goal. This cost for chart  $\mathcal{C}_i$  is heuristically evaluated as

$$h_i = \beta^{n_i} \|\mathbf{x}_i - \mathbf{x}_g\|, \quad (16)$$

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

Observe that, in contrast with  $A^*$ , the Greedy Best-First search, does not necessary generate all the neighbors of a chart under expansion. The generation of children charts proceeds only while the children have higher cost than the parent. This strongly reduces the final number of charts.

Finally, note that the path obtained by the Greedy Best-First search is not necessarily optimal. As mentioned, the generation of a (resolution) optimal path would require an  $A^*$  search, which, in practice, is too expensive.

## 4.2 Chart expansion: Generating random directions

While the chart to extend is selected greedily, the exact direction of expansion is selected randomly, allowing the discovery of alternative paths in the presence of obstacles. The random selection of expansion points is implemented by sampling a point uniformly on the surface of the ball associated with the chart and checking if this point is inside the corresponding polytope. If it is the case, the generation of the new chart proceeds as detailed in Section 3. Otherwise, no chart is generated since points on the ball surface but outside the corresponding polytope are in areas covered by neighboring charts.

The uniform generation of random points,  $\mathbf{u}_i^j$ , on the surface of the  $k$ -dimensional ball of chart  $\mathcal{C}_i$  is done by generating each element of  $\mathbf{u}_i^j$  according to a normalized one-dimensional Gaussian and scaling the resulting vector to norm  $r$  (Fishman, 1996).

To check if the point is inside the associated chart's polytope, we exploit the fact that convex polytopes can be represented as the intersection of  $k$ -dimensional half-spaces given by the polytope faces (Chen et al., 1991). Thus, for a point  $\mathbf{u}_i^j = (u_1, \dots, u_k)^\top$  to be inside the polytope  $\mathcal{P}_i$  made of  $m_i$  faces, it must fulfill

$$\gamma_0^t + \sum_{w=1}^k \gamma_w^t u_w \geq 0, \quad (17)$$

for all the faces  $f^t = (\gamma_0^t, \dots, \gamma_k^t)^\top$ ,  $t = 1, \dots, m_i$  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 the corresponding mapping  $\psi$ . Then we check whether the projected point is close to the current chart, whether the curvature of the tangent space at this point is similar to that of the current chart and whether the point is collision free. The last valid configuration is used as a center for a new chart. This adjusts the distribution of charts not only to the local manifold curvature but also to the shape of the collision-free configuration space. If no progress at all

---

### Algorithm 1: HC-path planner.

---

**HC-Planer**( $\mathbf{x}_s, \mathbf{x}_g, \mathbf{F}$ )  
**input** : A couple of samples to connect,  $\mathbf{x}_s$  and  $\mathbf{x}_g$ , and a set of constraints,  $\mathbf{F}$ .  
**output**: A path connecting the two samples.

- 1  $\mathcal{C}_s \leftarrow \text{INITCHART}(\mathbf{x}_s, \mathbf{F})$
- 2  $\mathcal{C}_g \leftarrow \text{INITCHART}(\mathbf{x}_g, \mathbf{F})$
- 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  $\mathcal{C}_i \leftarrow \text{EXTRACTMIN}(H)$
- 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  $A \leftarrow A \cup \{\mathcal{C}_j\} \cup \text{BIFCHARTS}(\mathcal{C}_i, \mathcal{C}_j)$
- 15  $h_j \leftarrow \|\mathbf{x}_j - \mathbf{x}_g\|$
- 16  $H \leftarrow \text{ADDTOHEAP}(\mathcal{C}_j, h_j)$
- 17 **RETURN**(PATH( $A, \mathcal{C}_s, \mathcal{C}_g$ ))

---

can be done towards the target point, the expansion is declared as failure and the chart under expansion is penalized. Since no check is done between intermediate points,  $\delta$  has to be set small enough so that only minor interpenetrations and curvature changes could occur.

Once the goal configuration is reached, the path to the start is reconstructed using the parent-children charts relations.

## 4.3 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 goal configurations respectively, and tries to connect them with a path on the manifold 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 and 2). Each chart is a tuple  $\mathcal{C} = \{\mathbf{x}, \Phi, \mathcal{B}, \mathcal{P}\}$  where  $\mathbf{x}$  is the center of the chart,  $\Phi$  is the orthonormal base of the tangent space of the manifold at  $\mathbf{x}$ , and  $\mathcal{B}$  and  $\mathcal{P}$  are, respectively, the ball and the polytope bounding the area of applicability of the chart, both defined in the tangent space. The two charts are then included in the initial atlas,  $A$  (line 3). To efficiently determine the chart with the minimum expected cost, charts are organized into a binary heap. Thus, the heuristic-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

---

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

---

**GenerateNewChart**( $\mathcal{C}_i, \mathbf{F}$ )**input** : A chart to expand,  $\mathcal{C}_i$ , 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    $d \leftarrow \delta$ 
8   while  $d \leq r$  and  $e$  and  $c$  and  $t$  do
9      $\hat{\mathbf{x}}_i^j \leftarrow \mathbf{x}_i + \Phi_i(d/r) \mathbf{u}_i^j$ 
10     $\mathbf{x}_j \leftarrow \text{PROJECT}(\mathcal{C}_i, \hat{\mathbf{x}}_i^j, \mathbf{F})$ 
11    if  $\|\hat{\mathbf{x}}_i^j - \mathbf{x}_j\| > \sigma$  then
12       $e \leftarrow \text{FALSE}$ 
13    else
14       $\Phi_j \leftarrow \text{TANGENTSPACE}(\mathbf{x}_j, \mathbf{F})$ 
15      if  $\|\Phi_i^\top \Phi_j\| < 1 - \sigma$  then
16         $t \leftarrow \text{FALSE}$ 
17      else
18        if  $\text{COLLISION}(\mathbf{x}_j)$  then
19           $c \leftarrow \text{FALSE}$ 
20        else
21           $\mathcal{C}_j \leftarrow \text{INITCHART}(\mathbf{x}_j, \mathbf{F})$ 
22           $d \leftarrow d + \delta$ 
23 RETURN( $\mathcal{C}_j$ )
```

---

search is performed as described in Section 4.1, 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 of this chart,  $\mathcal{P}_i$ , still has vertexes outside the ball  $\mathcal{B}_i$  (line 8), we try to extend the atlas with a new chart (line 9). If the extension fails (line 10), the current chart is penalized (line 11) so that its chances to be selected for future extension decreases, and it is returned 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. Every time a new chart is added to the atlas, we check whether the line connecting the centers of the parent and the child charts crosses a bifurcation and, if so, we generate charts modeling it, using the numerical procedure described at the end of Section 3.3. Function `BIFCHARTS` (line 14) implements this process and returns the new charts or an empty set if there is no bifurcation between  $\mathcal{C}_i$  and  $\mathcal{C}_j$ . Finally, the heuristic-to-goal is initialized for the new chart (line 15) and added to the heap (line 16). When the goal is reached, the parent-child

relations between charts can be used to extract the path linking the query configurations via the centers of some of the charts in the atlas. Note that the atlas is divided in two sets, the one containing  $\mathbf{x}_s$  and the one including  $\mathbf{x}_g$  and that in the presented algorithm only the first one is extended. A bidirectional search could be implemented alternatively extending the two parts of the atlas. This variant could be very effective for some obstacle arrangements, but its description is out of the scope of this paper.

The generation of a new chart from a previous one is presented in Algorithm 2. We select a point  $\mathbf{u}_i^j$  on surface of the ball defined on the tangent space of the input chart (line 2), as described in Section 4.2. If the point is inside the polytope (line 3), i.e., the point is in the potential area of applicability of the current chart, we proceed to determine a point,  $\mathbf{x}_j$ , adequate to generate a new chart. This point is searched from the center of the chart under expansion, progressively 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 and 10) using Eqs. (4) and (5) that implement the mapping  $\psi_i$ . If the projection converges to a point on the manifold, we check whether the obtained point is too far away from the tangent space (line 11), whether the tangent space at the new point, computed using Eq. (2), and that of  $\mathcal{C}_i$  are too different (line 15), and whether it is in collision (line 18). In any of these cases, the progress towards the new point is stopped (lines 12, 16, and 19) and we return the chart based on 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  $\mathbf{F}$ , a  $(n - k) \times n$  matrix, and we have to define a box with  $2^k$  vertexes. The initialization of the heap is  $O(1)$  and its query (i.e., the extraction and removal of its minimum element) is  $O(k)$  since it is logarithmic with the number of elements in the heap (i.e., the number of charts in the atlas) that is exponential in  $k$ . The cost of the six basic steps involved in the chart generation is the following: 1) the generation of a random sample on a  $k$ -dimensional ball (line 2) is  $O(k)$ ; 2) the test to determine if a point is inside a  $k$ -dimensional polytope (line 3) scales as  $O(k 2^k)$  since each face is defined by a  $k + 1$  dimensional vector and the number of neighbors of a chart grows with the kissing number that is  $O(2^k)$ ; 3) the projection of a point from the tangent space to the manifold (lines 9 and 10) is  $O(n^3)$  since it is implemented as a Newton process with a bounded number of iterations where at each iteration a QR decomposition is used; 4) the generation of the

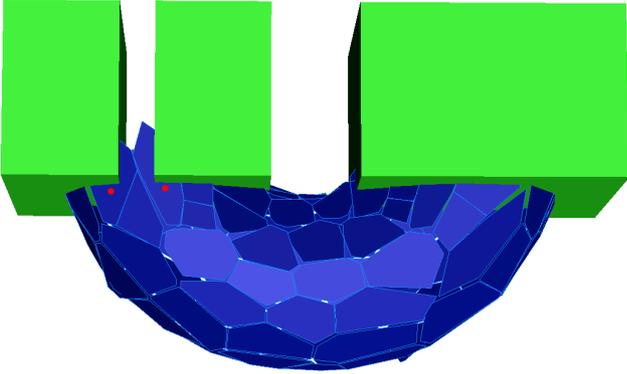


Figure 6: A torus-like configuration space with obstacles and a narrow corridor. The green blocks are obstacles and the charts are depicted in blue. When trying to enter the narrow corridor using too large charts, the algorithm can get blocked. There is not straight line between the center of the charts at the entrance of the corridor (marked with red dots) and the polytope vertices that can be used to expand the atlas. In this case, the width of the corridor is 0.125 and the radius of the charts is 0.5. The deadlock is avoided with radius below 0.0625 although larger radius can also be used in practice due to the randomness in the algorithm.

tangent space to the manifold at  $\mathbf{x}_j$  (line 14) scales as  $O(n^3)$ ; 5) the determinant of a matrix of size  $k \times k$  that comes from the product of  $n \times k$  matrices (line 15) is  $O(k^3 + k^2 n)$  and, finally; 6) the initialization of a chart (line 21) is  $O(2^k)$ , since the tangent space computed at step 4 is reused. The detection of possible bifurcations between the new chart and its parent basically scales with  $O(n^3)$  since the dicotomic search requires the evaluation the determinant of a  $n \times n$  matrix which is implemented via LU decomposition. The rest of the operations in the bifurcation detection are cheaper than this. When adding a chart to the atlas, we have to look for neighboring charts. The selection of potential neighbors can be done in  $O(k)$  since, using a kd-tree, this operation is logarithmic with the number of charts that, in the worst case, scales exponentially with  $k$ . For each potential neighbor we have to determine if its center is close enough to the new chart and whether or not the tangent spaces are similar. These two steps are equivalent to steps 3 and 5 described before and, thus, they scale as  $O(n^3)$  and  $O(k^2 n + k^3)$ , respectively. For the actual neighboring charts, we have to crop the corresponding polytopes. This operation scales with the number of vertexes of those polytope which is  $O(2^k)$ . Finally, the addition of an element to the heap is  $O(k)$ , in agreement with the cost updating it.

Summarizing, in highly constrained systems as the

ones we address and where  $k \ll n$ , the cost of the overall algorithm scales with  $O(l n^3)$  where  $l$  is the number of charts needed to connect the start and the goal. The number of charts  $l$  dominates the cost and depends on the complexity of the manifold to trace out. In the worst case, the final atlas might include all the possible charts for a given manifold and, then,  $l$  is exponential in  $k$  and independent of  $n$ . However, as we show in next section, many problems require in practice a limited number of charts to connect the start and goal configurations.

Despite randomness allows to benefit from an infinite set of directions for the expansion of the atlas, the probabilistic completeness of the planner cannot be guaranteed. The expansions always occur from the center of a chart and in some situations all possible expansion directions can be blocked by obstacles, as illustrated in Fig. 6. Note that such a case is unlikely and it can always be avoided using a small enough  $r$ . Therefore, the planner is resolution complete, in the sense that by taking a radius  $r$  small enough we can ensure to find a solution path if it exists. In particular, in problems involving narrow passages of minimum width  $v$ , setting  $r < v/2$  would ensure a solution. However, in practice, much larger radius can be used safely.

## 5 Experiments

The HC-planner described in Section 4 based on the higher-dimensional continuation tools described in Section 3 including the treatment of bifurcations was implemented in C.<sup>1</sup> These tools were integrated as modules of our position analysis toolbox (KRD Group, 2011) using SOLID (van den Bergen, 2007) as a collision detector, and the GNU Scientific Library (Galassi et al., 2009) for the lineal algebra operations. The position analysis toolbox is based on a formulation with redundant variables that yields a system of equations only containing linear, bilinear, and quadratic monomials (Porta et al., 2009). This offers the possibility of testing the algorithms in challenging situations where the dimensionality of the configuration space is much lower than that of the ambient space. Moreover, using a redundant formulation, the configuration spaces tend to be less convoluted, favoring both the projection of samples from the ambient space to the configuration space and the trace of this space using continuation-based methods (Wampler and Morgan, 1991). Note that the system introduced by Um et al. (2010) is not suitable for redundant formulations. Thus, for the purpose of comparison, we implemented the Constrained-Connect RRT (CC-

<sup>1</sup>An implementation of the higher-dimensional continuation tools tailored for Dynamical Systems is offered by Henderson (2002b).

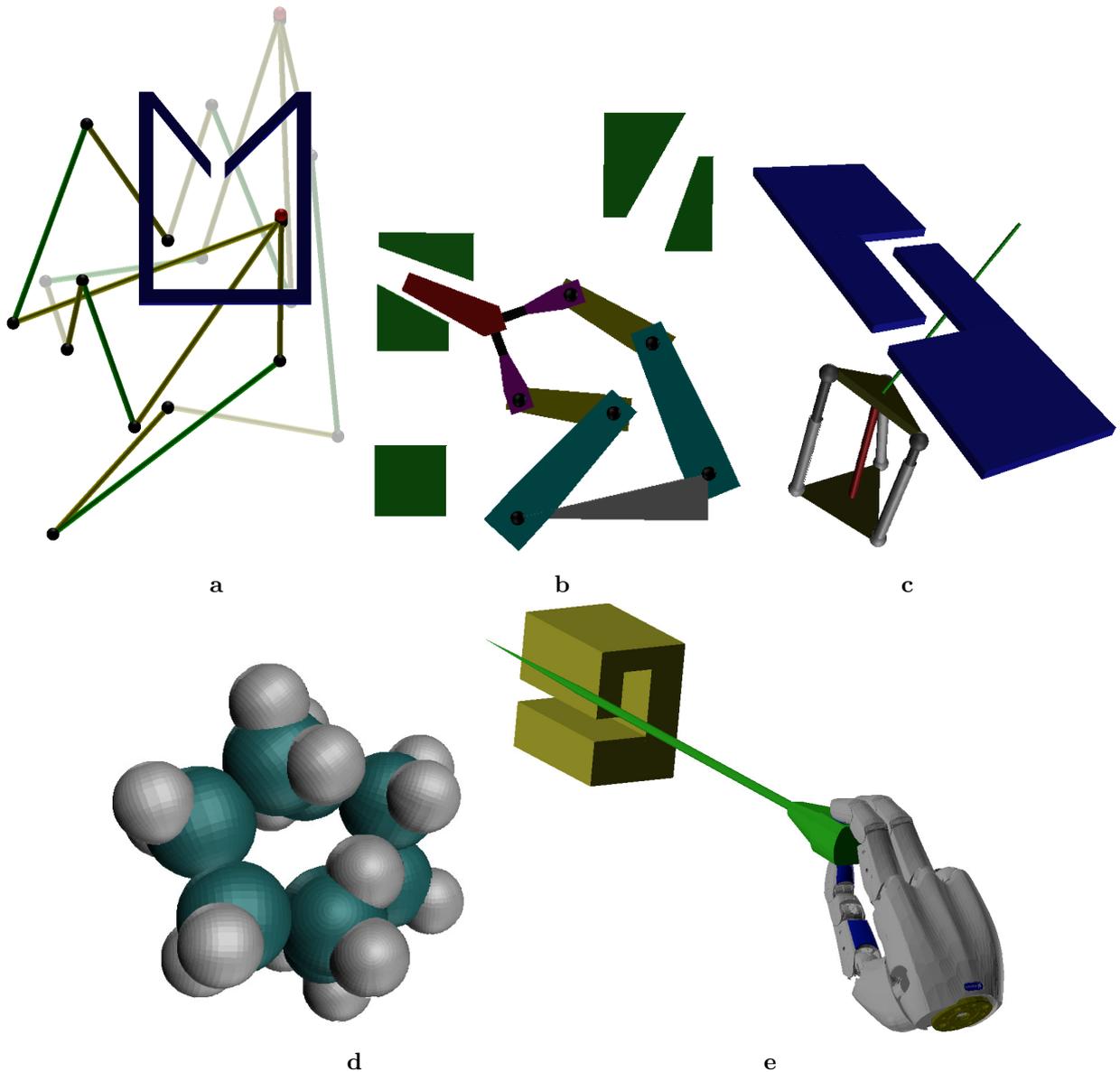


Figure 7: The five benchmarks used in the experiments. **a** A star-shaped planar manipulator with three fixed extremities and a bug trap like obstacle. **b** A two-arms manipulator moving an object from one gap to another. **c** A rotational-only parallel manipulator moving a stick among some obstacles. **d** The cyclooctane molecule. **e** The Schunk anthropomorphic hand performing an insertion task.

RRT), a recent method for planning in constrained spaces introduced by Dalibard et al. (2009) that, like the HC-planner, implements an unidirectional search and where several nodes are added at each tree extension. In the CC-RRT, points are sampled in the ambient space and the nearest sample on the manifold is progressively extended towards the random sample with small steps of size  $\delta$  along the line connecting the two points. At each extension step, the intermediate points are projected to the manifold using the Jaco-

bian pseudo inverse method. In our implementation, the CC-RRT nearest-neighbor queries use the kd-tree described by Yershova and LaValle (2007) and all executions are limited to a maximum of 600 seconds. Experiments were executed on a Intel Core i7 at 2.93 Ghz running Mac OS with parameters  $r = 0.4$ ,  $\delta = 0.05$ ,  $\sigma = 0.1$ , and  $\beta = 1.1$ .

Figure 7 shows the five benchmarks used in this paper. The first one is a planar star-shaped manipulator also used by Shvlab et al. (2007), but adding obsta-

Benchmarks	$k$	$n$	CC-RRT		HC-planner		CC-RRT/HC	
			Time	Samples	Time	Charts	T/T	S/C
Star-shaped	5	18	257.06	219658	22.62	5015	11.46	43.80
Two-arms	3	10	37.89	40554	0.17	412	222.88	98.43
Parallel	3	27	17.33	9089	0.28	184	61.89	49.39
Cyclooctane	2	17	2.69	4486	0.11	138	24.45	32.50
Schunk hand	2	177	-	-	68.52	59	-	-

Table 1: Dimensionality of the configuration and ambient spaces, execution times and number of samples/charts used by the CC-RRT and the HC-planner. The CC-RRT is not able to solve the Schunk hand problem in less than 600 seconds.

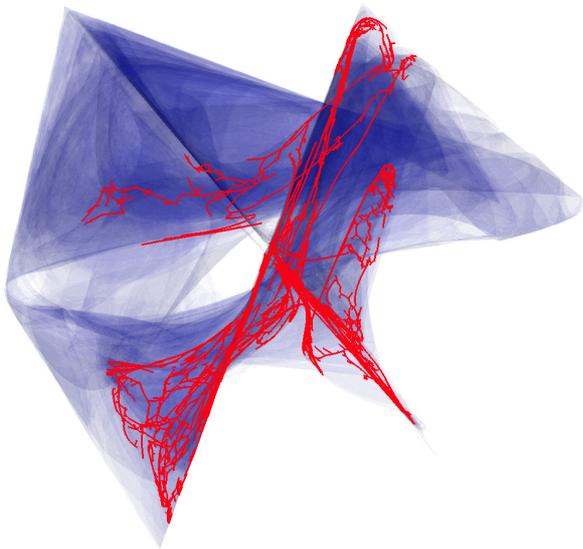


Figure 8: Full atlas for the Schunk hand benchmark projected in three variables (in translucent blue). The atlas is build without considering joint limits nor obstacles and using a CC-RRT with 75000 nodes (red lines) departing from a given sample an trying to cover the whole configuration space. The CC-RRT has difficulties in covering the manifold due to its intricate structure.

cles forming a bug trap to evaluate the performance of the algorithms in the presence of local minima. Collisions are only considered between the bug trap and the point where the three legs coincide (shown in red in the figure). The goal configuration for this example is shown in translucent colors in the figure. In this particular experiment, parameters  $r$  and  $\sigma$  are enlarged to 1.5 and 0.5, respectively, to reduce the number of charts to generate due to the high dimensionality of the configuration space. 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 (Gharbi et al., 2008). The move-

ment between the start and goal configurations requires to traverse singularities for the individual manipulators<sup>2</sup>, which makes the problem unsolvable by basic Kinematics-based Roadmap approaches (Cortés et al., 2002; Han and Amato, 2000). 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 through a gap among some obstacles. The fourth benchmark is the cyclooctane, a molecule whose kinematics is a 8-revolute loop. 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). The last benchmark is an insertion task with the Schunk anthropomorphic hand (Schunk GmbH & Co. KG, 2006). In this task, the hand is grasping a needle that has to be inserted into a gap in an object. The relative position of the hand palm and the object is fixed and, thus, the insertion has to be done moving the fingers and without releasing the needle. This example is specially remarkable since using our redundant formulation (Rosales et al., 2011) the dimensionality of the ambient space is much larger than that of the configuration space (177 versus 2, respectively), demonstrating the scalability of the HC-planner.

Table 1 shows the performance comparison between CC-RRT and the HC-planner averaged over 100 runs. For each of the five 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 samples or charts used by each method. The table also shows execution time ratios (T/T) as well as the ratio between the number samples used in CC-RRT and the number of charts used with the HC-planner (S/C).

The results of the experiments show that, for this set of problems, the execution time of the CC-RRT is at least one order of magnitude higher than the one

<sup>2</sup>Those singularities are not the same as the singularities described in Section 3.3. Here the Jacobian of passive variables losses rank, but not the full system Jacobian.

Benchmarks	C-Bi-RRT		HC-planner		C-Bi-RRT/HC	
	Time	Samples	Time	Charts	T/T	S/C
Star-shaped	38.02	80116	22.62	5015	1.68	15.97
Two-arms	0.92	2867	0.17	412	5.41	6.95
Parallel	11.70	1852	0.28	184	41.78	10.06
Cyclooctane	0.67	1510	0.11	138	6.09	10.94
Schunk hand	-	-	68.52	59	-	-

Table 2: Execution times and number of samples/charts used by the C-Bi-RRT and the HC-planner.

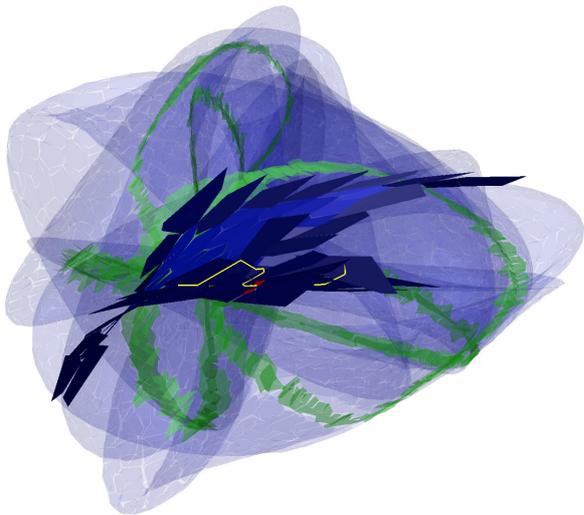


Figure 9: Full atlas of the configuration space of the cyclooctane (in translucent blue) and partial atlas necessary to determine a path between two configurations (in solid blue) projected in three variables. The solution path is shown as a yellow line. Only a small part of all possible charts are necessary to solve the path planning problem. Charts in translucent green are generated at bifurcations as described in Section 3.3. In this case, the solution path crosses the bifurcation via the red chart in the center of the figure.

of the HC-planner This is true despite generating samples is much faster than generating charts. However, charts are more powerful since they do not only describe the manifold on a single point but on a local neighborhood of a point. Thus, the HC-algorithm uses in average 50 times less charts than samples used by CC-RRT. In general, as the dimensionality of the configuration space,  $k$ , increases the number of charts to generate significantly grows and, thus, the HC-planner becomes less efficient. Despite this, in the star-shaped example where  $k = 5$  the HC-planner is still remarkably more efficient than the CC-RRT. On the other hand, as the dimensionality of the ambient space,  $n$ , increases the cost of generating each chart increases

too. However, even for  $n = 177$  the HC-planner correctly solves the problem in all cases, while CC-RRT is not able to find a solution in less than 600 seconds.

As any strategy based on sampling in the ambient space, the CC-RRT is affected by an inadequate sampling bias. This problem is highlighted in Fig. 8. This figure shows the projection on three of the variables of the problem of the full atlas of the configuration space for the Schunk hand example but without considering joint limits nor obstacles. As it can be seen, the surface defining the configuration space is heavily folded and, therefore there is no guarantee that samples projected from the ambient space will uniformly cover the configuration space surface. This complicates the exploration of this space using CC-RRT. The figure shows a tree built on this manifold by CC-RRT from a given sample, without any specific goal (i.e., trying to cover the whole configuration space), and using 75000 nodes. It can be appreciated that despite the large number of samples, the tree is far from covering the whole configuration space and it does not exhibit the typical Voronoi bias of RRTs when exploring Euclidean spaces (LaValle and Kuffner, 2000). Since in this experiment obstacles are not present, the inadequate bias can only be attributed to the constraints defining the manifold. Using a non-redundant formulation, the configuration space is likely to be even more convoluted, hindering even more its exploration from samples in the ambient space.

The relevance of dealing with bifurcations is shown in Fig. 9. This figure reports on a particular experiment with the cyclooctane where the path between the initial and the final configurations necessarily traverses a bifurcation switching from one branch of the manifold to another. The bifurcation set of the molecule is shown in translucent green in the figure and the particular point at which the bifurcation is crossed is shown as small red patch in the center of the figure. If bifurcations were not considered, it would be impossible to find the path connecting the query configurations. In this experiment, the partial atlas necessary to solve the path planning problem includes only 200 charts while the full atlas includes about 6000 charts.

The advantage of the HC-planner over CC-RRT

comes either from a more efficient tree extensions over the manifold or from the bias to the goal due to the use of the best-first search strategy. To determine which one of these two factors is more relevant, we compared the HC-planner with the C-Bi-RRT planner for constrained spaces proposed by Berenson et al. (2009, 2011). This planner uses a RRT-connect strategy (although different from that in the CC-RRT) and a bidirectional search which introduces a bias towards the goal similar to best-first strategy used in the HC-planner. Table 2 compares C-Bi-RRT and the HC-planner for the five benchmarks used in this paper. It can be appreciated that due to the bidirectional search, C-Bi-RRT is significantly more efficient than CC-RRT, but is still slower than the HC-planner. The star-shaped example with the bug trap obstacles is a remarkable case since, in principle, this type of obstacle arrangements are well suited for bidirectional search (LaValle, 2006). The HC-planner is still faster than the C-Bi-RRT, despite using a unidirectional search. Other problems might be devised where the C-Bi-RRT would outperform the HC-planner. Note, though, that a bidirectional version of the HC-planner would probably outperform C-Bi-RRT for those cases too. The advantage of the HC-planner over C-Bi-RRT confirms that a relevant part of the efficiency of the HC-planner comes from its more efficient coverage of the manifold and not only from the search strategy used.

The advantage of the HC-planner search strategy with respect to a more optimal strategy is evaluated by applying a standard A\* algorithm, implemented as described in the introduction of Section 4. In this case, the maximum number of charts in the atlas is set to 15000. With this setting, A\* is not able to solve the star-shaped problem. This is due to the curse of dimensionality: in a 5-dimensional configuration space the number of neighboring charts for each chart is about 40 and this results in a large exponential growth of the number of charts to generate. The two-arms benchmark can not be solved with  $r = 0.4$  since with such a coarse resolution, almost all the transitions between chart centers are in collision. With  $r = 0.25$  the problem can be solved, but with this resolution the number of charts to generate increases to the point that the A\* execution time is about 5 seconds. With  $r = 0.05$  the granularity is so thin that 15000 charts are not sufficient to connect the query configurations. Our approach dynamically adapts the center of the charts to the distribution of obstacles, avoiding the generation of charts on blocked regions and, thus, it can solve the problem without resorting to finer granularity. The parallel manipulator, the cyclooctane and the Schunk hand benchmarks can be solved with A\* in 17, 4, and 525 seconds, respectively, times significantly higher than the ones used by HC-planner. When suc-

cessful, A\* returns shorter paths than those obtained with the HC-planner. However, the path length for the HC-planner could be improved with a smoothing post-process, which is not yet implemented in our system. Finally, note that A\* is able to detect if two samples can not be connected because they are in different connected components of the manifold even in the presence of obstacles: in the worst case the atlas for the connected component including the start sample will be completed and if it does not include the goal sample the planning can be declared as a failure. The HC-planner trades off this completeness for efficiency.

## 6 Conclusions

In this paper, we extended the higher-dimensional continuation algorithmic tools to path planning applications. Relying on these tools, we defined a randomized path planner for highly constrained systems that directly works on the configuration space, trying to connect any pair of query configurations with a small collection of local charts. The algorithm performance in terms of coverage of the configuration space is highly independent of the relation between the configuration and the ambient spaces, which 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 significantly faster than the most recent alternative algorithms.

The worst case 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 (Canny, 1988). Thus, the algorithm would not scale gracefully to high-dimensional problems and, therefore, it is specially adequate for highly constrained system, i.e., systems where the dimension of the ambient space is much higher than that of the configuration space and where this space is of moderate dimension. Despite this worst case cost, the use of a greedy search strategy and the randomization allow to solve problems with moderate complexity (at least up to dimension 5 in the presented experiments) embedded in even higher-dimensional spaces (up to 177 in the experiments). This level of complexity includes many interesting problems in Robotics and in Molecular Biology (Brown et al., 2008). To scale to problems with even larger dimensionality we could rely on charts with larger applicability areas. However, the use of large charts limits the set of problems that can be solved since environments densely populated with obstacles typically require small charts. We would like to explore the possibility to define variants of the HC-planner where the role of the atlas granularity is minimized in the same way as probabilistic roadmaps overcome the resolution limitations of approximate cell

decomposition methods. Another possibility to explore is to define a cost function over the configuration space so that the exploration could be limited to areas with low cost (Jaillet et al., 2010). All these points deserve a more careful evaluation. In the experiments presented in this paper we used a redundant formulation since, in principle, it defines less convoluted configuration space manifolds. However, redundant formulations result in large Jacobian matrices and this affects the performance of sampling-based and continuation methods. Our preliminary experiments show that both methods could benefit in the same proportion form using a non-redundant formulation, but a more thorough testing is necessary to confirm these results. Finally, it is also our future endeavor to explore the possible extension of the proposed planner to problems with differential constraints.

## Acknowledgments

We would like to thank L. Ros for his support during the elaboration of this work. This work has been partially supported by the Spanish Ministry of Science and Innovation under project DPI2010-18449.

## References

- Ballantyne, G. and Moll, F. (2003). The da Vinci telerobotic surgical system: Virtual operative field and telepresence surgery. *Surgical Clinics of North America*, 83(6):1293–1304.
- Berenson, D., Srinivasa, S., Ferguson, D., and Kuffner, J. (2009). Manipulation planning on constraint manifolds. In *IEEE International Conference on Robotics and Automation*, pages 1383–1390.
- Berenson, D., Srinivasa, S., and Kuffner, J. (2011). Task space regions: A framework for pose-constrained manipulation planning. *International Journal of Robotics Research*, 30(12):1435–1460.
- Beyn, W.-J., Champneys, A., Doedel, E., Govarets, W., Kuznetsov, U. A., Yu, A., and Sandstede, B. (2002). *Handbook of Dynamical Systems (Vol 2)*, chapter Numerical Continuation, and Computation of Normal Forms, pages 149–219. Elsevier.
- Bohigas, O. (2011). Branch switching from singular points. Institut de Robòtica i Infomàtica. CSIC-UPC. Technical Report.
- Brown, W. M., Martin, S., Pollock, S. N., Coutsiaris, E. A., and Watson, J.-P. (2008). Algorithmic dimensionality reduction for molecular structure analysis. *Journal of Chemical Physics*, 129(6):064118.
- Canny, J. (1988). *The Complexity of Robot Motion Planing*. MIT Press.
- Chen, P.-C., Hansen, P., and Jaumard, B. (1991). On-line and off-line vertex enumeration by adjacency lists. *Operation Research Letters*, 10:403–409.
- Cortés, J., Siméon, T., and Laumond, J.-P. (2002). 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.
- Dalibard, S., Nakhaei, A., Lamiriaux, F., and Laumond, J.-P. (2009). 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.
- Fishman, G. F. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer.
- Galassi et al., M. (2009). *GNU Scientific Library Reference Manual*. Network Theory Ltd.
- Gharbi, M., Cortés, J., and Siméon, T. (2008). A sampling-based path planner for dual-arm manipulation. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 383–388.
- Han, L. and Amato, N. M. (2000). A kinematics-based probabilistic roadmap method for closed chain systems. In *Algorithmic and Computational Robotics - New Directions (WAFR2000)*, pages 233–246.
- Han, L., Rudolph, L., Blumenthal, J., and Valodzin, I. (2008). Convexly stratified deformation spaces and efficient path planning for planar closed chains with revolute joints. *International Journal of Robotics Research*, 27(11-12):1189–1212.
- Havoutis, I. and Ramamoorthy, S. (2009). 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.
- Henderson, M. E. (2002a). Multiple parameter continuation: Computing implicitly defined k-manifolds. *International Journal of Bifurcation and Chaos*, 12(3):451–476.
- Henderson, M. E. (2002b). The MultiFario project web page. <http://multifario.sourceforge.net>.
- Henderson, M. E. (2005). Multiparameter parallel search branch switching. *International Journal of Bifurcation and Chaos in Applied Science and Engineering*, 15(3):967–974.

- Henderson, M. E. (2007). *Numerical continuation methods for dynamical systems: path following and boundary value problems*, chapter Higher-Dimensional Continuation. Springer.
- Jaillet, L., Cortés, J., and Siméon, T. (2010). Sampling-based path planning on configuration-space costmaps. *IEEE Transactions on Robotics*, 26(4):635–646.
- Kavraki, L. E., Svestka, P., Latombe, J.-C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.
- Keller, H. B. (1977). *Applications of Bifurcation Theory*, chapter Numerical Solution of Bifurcation and Nonlinear Eigenvalue Problems. Academic Press.
- Krauskopf, B., Osinga, H. M., and Galán-Vioque, J. (2007). *Numerical continuation methods for dynamical systems: path following and boundary value problems*. Springer.
- KRD Group (2011). The CUIK project web page. <http://www.iri.upc.edu/research/webprojects/cuikweb>.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press, New York.
- Lavalle, S. M. (2011). Motion planning. Part I: The essentials. *IEEE Robotics and Automation Magazine*, 18(1):79–89.
- LaValle, S. M. and Kuffner, J. (2000). Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics - New Directions (WAFR2000)*, pages 293–308.
- Ott, C., Eiberger, O., Friedl, W., Bauml, B., Hillenbrand, U., Borst, C., Albu-Schafer, A., Brunner, B., Hirschmuller, H., and Hirzinger, G. (2006). A humanoid two-arm system for dexterous manipulation. In *IEEE-RAS International Conference on Humanoid Robots*, pages 276–283.
- Porta, J. M. and Jaillet, L. (2010). Path planning on manifolds using randomized higher-dimensional continuation. *Workshop on the Algorithmic Foundations of Robotics*, pages 337–353.
- Porta, J. M., Ros, L., and Thomas, F. (2009). A linear relaxation technique for the position analysis of multiloop linkages. *IEEE Transactions on Robotics*, 25(2):225–239.
- Pressley, A. (2001). *Elementary Differential Geometry*. Springer Verlag.
- Rheinboldt, W. C. (1988). On the computation of multi-dimensional solution manifolds of parameterized equations. *Numerische Mathematik*, 53:165–181.
- Rheinboldt, W. C. (1996). MANPACK: A set of algorithms of computations on implicitly defined manifolds. *Computers and Mathematics with Applications*, 32(12):15–28.
- Rodríguez, A., Basañez, L., and Celaya, E. (2008). A relational positioning methodology for robot task specification and execution. *IEEE Transactions on Robotics*, 24(3):600–611.
- Rosales, C., Ros, L., Porta, J. M., and Suárez, R. (2011). Synthesizing grasp configurations with specified contact regions. *International Journal of Robotics Research*, 30(4):431–443.
- Roth, B. and Freudenstein, F. (1963). Synthesis of path-generating mechanisms by numerical methods. *ASME Journal of Engineering for Industry*, 85:298–307.
- Schunk GmbH & Co. KG (2006). Schunk anthropomorphic hand. <http://www.schunk.com>.
- Shkolmik, A. and Tedrake, R. (2009). Path planning in 1000+ dimensions using a task-space Voronoi bias. In *IEEE International Conference on Robotics and Automation*, pages 2892–2898.
- Shvlab, N., Liu, G., Shoham, M., and Trinkle, J. C. (2007). Motion planning for a class of planar closed-chain manipulators. *International Journal of Robotics Research*, 26(5):457–473.
- Sommese, A. J. and Wampler, C. W. (2005). *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific.
- Stilman, M. (2007). Task constrained motion planning in robot joint space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3074–3081.
- Stilman, M. (2010). Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics*, 26(3):576–584.
- Tang, X., Thomas, S., Coleman, P., and Amato, N. M. (2010). Reachable distance space: Efficient sampling-based planning for spatially constrained systems. *International Journal of Robotics Research*, 29(7):916–934.
- Tsai, L.-W. (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. John Wiley and Sons.

- Um, T. T., Kim, B., Suh, C., and Park, F. C. (2010). Tangent space RRT with lazy projection: An efficient planning algorithm for constrained motions. In *Advances in Robot Kinematics*, pages 251–260.
- van den Bergen, G. (2007). The SOLID web page. <http://www.dtecta.com>.
- Wampler, C. W. and Morgan, A. (1991). Solving the 6R inverse position problem using a generic-case solution methodology. *Mechanism and Machine Theory*, 26(1):91–106.
- Wedemeyer, W. J. and Scheraga, H. (1999). Exact analytical loop closure in proteins using polynomial equations. *Journal of Computational Chemistry*, 20(8):819–844.
- Whitney, D. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10:47–53.
- Yakey, J. H., LaValle, S. M., and Kavraki, L. E. (2001). Randomized path planning for linkages with closed kinematic chains. *IEEE Transactions on Robotics and Automation*, 17(6):951–959.
- Yang, F.-C. and Haug, E. J. (1994). Numerical analysis of the kinematic dexterity of mechanisms. *Journal of Mechanical Design*, 116:119–126.
- Yao, Z. and Gupta, K. (2005). 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.
- Yershova, A. and LaValle, S. M. (2007). Improving motion planning algorithms by efficient nearest neighbor searching. *IEEE Transactions on Robotics*, 23(1):151–157.
- Yershova, A. and LaValle, S. M. (2009). Motion planning for highly constrained spaces. In *Robot Motion and Control. Lecture Notes on Control and Information Sciences*, volume 396, pages 297–306.