Path Planning for Active Tensegrity Structures

Josep M. Porta*, Sergi Hernández-Juan

Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens i Artigas 4-6, 08028, Barcelona, Spain

Abstract

This paper presents a path planning method for actuated tensegrity structures with quasi-static motion. The valid configurations for such structures lay on an equilibrium manifold, which is implicitly defined by a set of kinematic and static constraints. The exploration of this manifold is difficult with standard methods due to the lack of a global parametrization. Thus, this paper proposes the use of techniques with roots in differential geometry to define an atlas, i.e., a set of coordinated local parameterizations of the equilibrium manifold. This atlas is exploited to define a rapidly-exploring random tree, which efficiently finds valid paths between configurations. However, these paths are typically long and jerky and, therefore, this paper also introduces a procedure to reduce their control effort. A variety of test cases are presented to empirically evaluate the proposed method.

Keywords: Tensegrity Structures, Path Planning, Differential Geometry, Higher-Dimensional Continuation.

1. Introduction

Tensegrity structures (Motro, 2003) are composed of compressed and tensioned elements assembled in equilibrium (see Fig. 1). These structures have many favorable properties such as scalability, energy efficiency, reliability, and a large stiffness-to-mass ratio (Skelton et al., 2001). Therefore, they have been used in a rich variety of fields including art (Snelson, web page), architecture (Hanaor, 1992), civil engineering (Korkmaz et al., 2010), telescope construction (Sultan et al., 1999), space applications (Tibert, 2003), flight simulators (Sultan et al., 2000), morphing wings (Moored and Bart-Smith, 2006), sensor design (Sultan and Skelton, 2004), and robot construction (Mirats-Tur and Camps-Sala, 2011). Moreover, natural structures such as cell cytoskeletons (Ingber, 1993), molecules (Liedl et al., 2010) or animal muscular systems (Moored et al., 2010) can be accurately modeled as tensegrity frameworks.

In many relevant applications, active elements are used to change the shape of the tensegrity structures (Sultan and Skelton, 2003; Paul et al., 2006; Samili and Motro, 2007; Tensegriteam, web page; Mirats-Tur and Camps-Sala, 2011; Rhode-Barbarigos

*Corresponding author at: Tel: +34 934015751. Fax: +34 934015750.
Email address: porta@iri.upc.edu (Josep M. Porta)

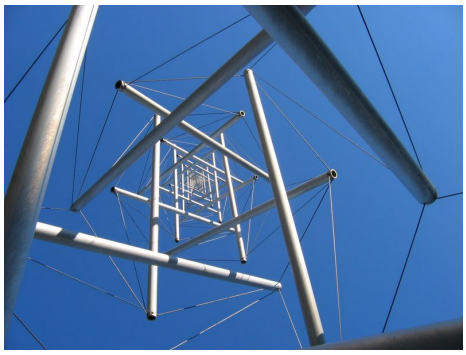


Figure 1: The Needle Tower, a tensegrity sculpture by Kenneth Snelson.

et al., 2012; Bruce et al., 2014). In general, moving quasi-statically is advantageous since it requires less powerful actuators. Thus, in most of the actual implementations, the shape changes are force-balanced at every step. Rigidity Theory (Connelly, 2013) rigorously describes the static equilibrium conditions for tensegrity structures. The assembly constraints together with these static conditions implicitly define an equilibrium manifold and the paths, i.e., flexes, between any two given configurations can be described as sequences of points on this manifold. Control techniques have been developed to track equilibrium paths (Pinaud et al., 2003; Wroldsen et al., 2009; Hernández-Juan et al., 2009), but the problem of efficiently defining such paths remains mainly open. This paper addresses this fundamental issue.

In Robotics, kinematic constraints also define a manifold that has to be explored to determine valid motions between configurations. Recently, novel numerical techniques with roots in differential geometry have been proposed to address this problem (Jaillet and Porta, 2011, 2013b). Herein these techniques are extended to apply them to tensegrity structures, where not only kinematic constraints have to be considered, but where, as described, static constraints also play a fundamental role.

This paper is organized as follows. First, Section 2 frames the contribution of this paper with respect to previous work. Then, Section 3 formulates the equations implicitly defining the equilibrium manifold, Section 4 describes the planning method proposed in this paper, Section 5 discusses implementation details of this method, Section 6 presents several experiments to evaluate it and, finally, Section 7 discusses the contributions and points to directions for further research.

2. Related Work

The problem of providing comprehensive characterizations of the equilibrium manifold of tensegrity structures is very challenging. Despite symmetries may increase the dimension of the manifold (Porta et al., 2013), they can also simplify the problem (Zhang et al., 2009b,a). For instance, in some symmetric structures it is possible to use analytical manipulation tools to obtain closed-form parameterizations of the equilibrium manifold (Sultan et al., 2001; Sultan and Skelton, 2003), which can be later exploited to design efficient deployment strategies (Sultan, 2014). Unfortunately, the approach does not generalize to arbitrary structures. The numerical methods presented herein can be used to represent the equilibrium manifold of general tensegrity structures defining an atlas, i.e., a set of coordinated charts, where each chart provides a

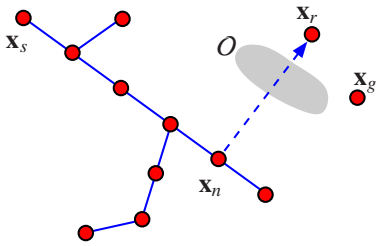


Figure 2: An RRT is built initializing an exploratory tree with $\mathbf{x}_s$. Branches are added to the tree generating a random configuration $\mathbf{x}_r$, determining $\mathbf{x}_n$, the node in the tree closer to $\mathbf{x}_r$, and creating a branch from $\mathbf{x}_n$ to $\mathbf{x}_r$. The branch is stopped if $\mathbf{x}_r$ is reached or if an obstacle region, O , is encountered. The tree extension continues until a branch is close enough to the goal configuration $\mathbf{x}_g$.

local parametrization of a given manifold. However, the computational cost of building an atlas rapidly grows with the dimension of the manifold.

Since the full description of the equilibrium manifold is difficult, partial information can be obtained by sampling it. This is the purpose of the form-finding methods (Tibert and Pellegrino, 2003a). The first form-finding methods were based on geometric reasoning and they can only be applied to particular structures (Fuller, 1962; Snelson, 1965). General solutions are typically based on iterative optimization processes. The difference between them is the function to optimize and the method used to optimize it. Analytical methods (Motro, 1984) try to find a valid configuration by maximizing/minimizing the length of struts/cables, but they are only effective on small or highly symmetric structures. Moreover, since these techniques focus on the kinematic parameters, they may converge to unstable configurations. Static methods overcome this issue by including the relation between the kinematic and the force parameters in the optimization. These methods can be based on energetic considerations (Connelly, 1982), on analytical procedures (Connelly and Terrell, 1995), on group theory (Murakami and Nishimura, 2001), on the rank of the force density matrix (Gómez-Estrada et al., 2006), on quasi-Newton methods (Hernández-Juan and Mirats-Tur, 2008), on finite element theory (Pagitz and Mirats-Tur, 2009), on genetic algorithms (Koohestani, 2012), or on combined equilibrium and geometrical compatibility equations (Koohestani and Guest, 2013), but in all cases they provide isolated points on the equilibrium manifold. An exception is the method proposed by Micheletti and Williams (2007) that generates sequences of valid configurations departing from a point already in equilibrium. However, this method, can not drive the structure to a specific goal nor take into account the presence of (self-)collisions along the flexes. Both aspects are properly considered herein extending techniques originally developed in Robotics.

The problem of finding collision-free paths connecting given configuration is a fundamental issue in Robotics. The most successful methods to address this problem are based on randomized sampling (LaValle, 2006) and, arguably, the most popular algorithm is the Rapidly-expanding Random Tree (RRT) (LaValle and Kuffner, 2000). This algorithm, illustrated in Fig. 2, defines an exploration tree from a start configuration $\mathbf{x}_s$. Branches are added to the tree until the goal configuration $\mathbf{x}_g$ is reached. Typically, the RRT branches are grown by linear interpolation between samples. However, this is only possible in spaces that are globally parameterizable and robots with kinematic constraints have manifold configuration spaces which, in general, do not have such parametrization. The usual alternatives to deal with such spaces are either intended for particular families of robots (Han and Amato, 2000; Han and Rudolph, 2006) or fail to efficiently explore the manifold (Berenson et al., 2011). To address theses is-

sues, Jaillet and Porta (2011) introduced the Atlas-based Rapidly-exploring Random Tree (AtlasRRT) planner. This planner implements higher-dimensional continuation tools (Henderson, 2002) to define an atlas over an implicitly-defined manifold departing form a point on it. During the definition of the atlas, the local parameterizations provided by the available charts are used to sample the manifold and to construct connections between the samples, defining an RRT. As the RRT grows, new charts are added to the atlas, if necessary. In the case of tensegrity structures, though, the manifold is not only defined by kinematic constraints, but also by static ones. Thus, this paper extends the AtlasRRT planner to deal with these latter constraints too. Up to our knowledge, the only approaches proposing RRT planners for tensegrity frameworks are (Hernández-Juan and Mirats-Tur, 2008) and (Xu et al., 2014). Both approaches use computationally expensive form-finding procedures to generate samples and to connect them. In contrast, the approach proposed herein is significantly more efficient since it approximates and exploits the structure of the underlying manifold. Moreover, none of the previous approaches have been proved to be complete, i.e., able to find a solution if it exist, while the approach presented herein is probabilistically complete, i.e., it will find the solution with probability one if it exists and enough time is granted to the planning process (Jaillet and Porta, 2013b).

RRT-based planners aggressively explore a given space, but the resulting paths are not optimal in any particular sense. In Robotics, several approaches have been proposed to approximate optimal paths either locally (Geraerts and Overmars, 2007) or globally (Karaman and Frazzoli, 2011). The latter approach integrates the optimization in the RRT construction and it may generate better paths, but the former is more efficient in practice. Herein, a local path optimization strategy is proposed to minimize the control effort, which is a measure particularly adequate for tensegrity structures.

3. Problem Formalization

A tensegrity structure defines an abstract tensegrity framework, i.e., a graph with q_e edges and q_n nodes. The edges are given by the tensegrity elements and the nodes by their connections. The set E of elements can be divided between a subset F of q_f elements with fixed length and a subset S of q_s stretchable elements. The first subset comprises the struts that only support compression and the second includes cables and springs that only exert tension. Bars are a special type of elements that can support compression or exert tension depending on the situation. In some cases, both struts and bars can be prismatic, and thus have variable length. Since the mass of struts and bars is large relative to the mass of cables and springs, the former are modeled as rigid elements and the latter are assumed to be massless elastic elements that follow Hooke's law. Additionally, the motions are assumed to be quasi-static, the structure is not supposed to interact with obstacles, but to avoid them, friction will be neglected, and, finally, external forces, if any, are assumed to be applied to the nodes.

When deriving the equations for a given abstract tensegrity framework we have to define the kinematic and static spaces where the configurations of the structure are represented and the corresponding kinematic and static constraints that determine the subset of valid configurations. While the kinematic constraints ensure that the structure is properly assembled, the static ones guarantee that the structure is in equilibrium. In this process, there is a trade off between compactness and simplicity. The traditional formulation based on director cosines (Sultan and Skelton, 2003) produces concise representations, but introduces trigonometric expressions that complicate the resulting

constraints. On the other hand, vector-based representations enlarge the space, but produce simple constraints, which leads to a simpler mathematical treatment (Williamson et al., 2003). This paper follows the second approach since it is in agreement with the philosophy of the CuikSuite (Porta et al., 2014), the software package in which we integrated the tools described herein.

Formally, the kinematic part of the configuration of a tensegrity structure will be represented by a vector $\mathbf{u}_j = (x_j, y_j, z_j)^\top \in \mathbb{R}^3$ for each $j \in E$ such that

$$\| \mathbf{u}_j \|^2 = 1, \tag{1}$$

defining the normalized direction vector of the element j of the tensegrity. Moreover, l_j will denote the constant or variable length of this element.

With this representation, an embedding of the abstract tensegrity framework (i.e., a set of coordinates for the nodes) is given by

$$\mathbf{p}_i = \mathbf{p}_0 + \sum_{j \in T_i} \sigma_j \, l_j \, \mathbf{u}_j, \tag{2}$$

for $i \in \{1, \ldots, q_n\}$, where $\mathbf{p}_0$ is a fixed position for an arbitrary node i_0, $T_i = \{j_1, \ldots, j_t\}$ are the tensegrity elements connecting i_0 and node i, σ_j is $+1$ or -1 depending on whether element j is traversed in the same orientation as $\mathbf{u}_j$, and l_j is the length of the corresponding element.

A kinematic configuration is valid if the different paths from i_0 to anyone of the remaining nodes give the same set of coordinates. This is ensured if, for all the cycles in a basis of cycles in the graph, we have that

$$\sum_{j \in C} \sigma_j \, l_j \, \mathbf{u}_j = \mathbf{0}, \tag{3}$$

with $C = \{j_1 \ldots . j_c\}$ the tensegrity elements forming the cycle (Koohestani and Guest, 2013).

The static part of the configuration of a tensegrity structure will be represented by a vector of forces, $\mathbf{f} = (f_1, \ldots, f_{q_e})^\top$, with one component for each element in the structure and where, by convention, compression and tension forces are assumed to be negative and positive, respectively. Since the stretchable elements follow Hook's law, we have that

$$f_j = k_j \, (l_j - r_j), \tag{4}$$

for $j \in S$, where k_j is the stiffness constant (negative/positive for compressive/tensile elements) that depends on the cross section area and the Young's modulus of the corresponding element, and r_j is its rest-length.

A static configuration is valid if at every node i the forces are balanced, i.e., if

$$\sum_{j \in A_i} \sigma_{i,j} \, f_j \, \mathbf{u}_j = \mathbf{f}_{e,i}, \tag{5}$$

where $A_i = \{j_1, \ldots, j_{a_i}\}$ are the tensegrity elements incident to node i, $\sigma_{i,j}$ is $+1$ if $\mathbf{u}_j$ is pointing to node i or -1 otherwise, and $\mathbf{f}_{e,i} \in \mathbb{R}^3$ is the external force applied to the node. Eq. (5) can be expressed for all nodes in matrix form

$$\mathbf{R}^\top \, \mathbf{f} = \mathbf{f}_e, \tag{6}$$

where $\mathbf{R}$ is a normalized rigidity matrix, i.e., a rigidity matrix (Connelly and Whiteley, 1992) with the entries divided by the length of the corresponding element, and where $\mathbf{f}_e = (\mathbf{f}_{e,1}^\top, \ldots, \mathbf{f}_{e,q_n}^\top)^\top$ is the vector of external forces.

Summarizing, the ambient space $\mathcal{A}$ where the configuration of a tensegrity structure is defined includes the three components of the normalized direction vector and the force for each tensegrity element and the length and rest-length for each stretchable element in the structure. The components of the director vectors can be bounded to the $[-1, 1]$ range. The compression and tension forces have negative and positive ranges respectively, their maximum magnitudes are given by the properties of the materials used to build the tensegrity structure, and their minimum magnitudes provide safety margins so that each element is properly stressed at any moment. Finally, the lengths of the elements are limited by the geometry of the structure, and the rest lengths can be bounded by Hook's law. Thus, since the ranges of all the components of the configuration are bounded, $\mathcal{A}$ is an axis-aligned box in $\mathbb{R}^m$, with $m = 4\,q_e + 2\,q_s$.

The set of equations implicitly defining the equilibrium manifold in $\mathcal{A}$ includes Eq. (1) for each element in the tensegrity, Eq. (3) for each cycle in a basis of cycles of the tensegrity graph, Eq. (4) for each stretchable element in the tensegrity structure, and the force balance constraints in Eq. (6). This defines a set of $n = 4\,q_e + q_s + 3$ equations. These equations allow trivial flexes such as rigid translations and rotations. If they are not desired, Eq. (2) can be used to set up the required tie-down constraints. Observe that m and n are potentially large, but the equations are simple since their degree is at most two.

The whole set of equations forms a non-linear system

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \tag{7}$$

with $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^n$, which implicitly defines the equilibrium set in $\mathcal{A}$

$$\mathcal{E} = \{\mathbf{x} \in \mathcal{A} \subset \mathbb{R}^m \mid \mathbf{F}(\mathbf{x}) = \mathbf{0}\}. \tag{8}$$

The formulation guarantees that $\mathbf{F}(\mathbf{x})$ is differentiable and we assume that its Jacobian is full rank for all $\mathbf{x} \in \mathcal{E}$, which is the generic situation according to Sard's theorem. Thus, $\mathcal{E}$ can be assumed to be a smooth k-dimensional manifold.

4. Path planning on the Equilibrium Manifold

Let $\mathcal{O}$ be the set of configurations in $\mathcal{E}$ that are in collision and $\mathcal{F} = \mathcal{E} \setminus \mathcal{O}$ the open set of the non-colliding configurations. Let also assume that $\mathbf{x}_s$ and $\mathbf{x}_g$ are both in $\mathcal{F}$. Then, the path planning problem consists in finding a collision free trajectory linking the query configurations while staying on the manifold i.e. to find a continuous function $\pi : [0, 1] \to \mathcal{F}$ with $\pi(0) = \mathbf{x}_s$, $\pi(1) = \mathbf{x}_g$. In practice the problem is discretized and the objective is to find a path, i.e., a dense sequence of points, $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ with $\mathbf{x}_i \in \mathcal{F}$, $\mathbf{x}_1 = \mathbf{x}_s$ and $\mathbf{x}_p = \mathbf{x}_g$.

To find solution paths over $\mathcal{F}$, we propose to define an exploration tree. First, Section 4.1 presents the basic procedure to trace paths over $\mathcal{F}$ exploiting the local parameterizations provided by a set of charts. Then, Section 4.2 describes how to use this basic procedure to define a Rapidly-exploring Random Tree (RRT) over $\mathcal{F}$ to find collision-free paths connecting arbitrary configurations. Finally, Section 4.3 complements the planner with a method to optimize the control effort of the resulting paths.

4.1. Tracing Paths on the Equilibrium Manifold

In differential geometry, a chart defines a local parametrization of a k-dimensional manifold around a given point $\mathbf{x}_i$ as a bijective map, $\mathbf{x}_j = \psi_i(\mathbf{t}_j)$, between parameters $\mathbf{t}_j \in \mathbb{R}^k$ and points $\mathbf{x}_j$ on the manifold, with $\psi_i(\mathbf{0}) = \mathbf{x}_i$. The map from the parameter space to the manifold is known as the exponential map and the inverse is the logarithmic map (do Carmo, 1976). Following Henderson (2002), these maps can be defined using the k-dimensional tangent space at $\mathbf{x}_i$. An orthonormal basis for this tangent space is given by the $m \times k$ matrix, $\mathbf{\Phi}_i$, satisfying

$$\begin{bmatrix} \mathbf{J}_i \\ \mathbf{\Phi}_i^\top \end{bmatrix} \mathbf{\Phi}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_k \end{bmatrix}, \tag{9}$$

with $\mathbf{J}_i$ the Jacobian of $\mathbf{F}$ evaluated at $\mathbf{x}_i$, and $\mathbf{I}_k$, the $k \times k$ identity matrix. The columns of $\mathbf{\Phi}_i$ are a basis of the infinitesimal motions that preserve the constraints in the problem. These motions are known as mechanisms in the context of tensegrity structures and as self-motions in Robotics. Using this basis, the exponential map ψ_i for a given set of parameters $\mathbf{t}_j \in \mathbb{R}^k$ is defined by first computing the ambient space coordinates

$$\hat{\mathbf{x}}_j = \mathbf{x}_i + \mathbf{\Phi}_i \, \mathbf{t}_j \, , \tag{10}$$

and then, orthogonally projecting $\hat{\mathbf{x}}_j$ to the manifold. This projection is the point $\mathbf{x}_j$ fulfilling

$$\begin{cases} \mathbf{F}(\mathbf{x}_j) = \mathbf{0}, \\ \mathbf{\Phi}_i^\top \, (\mathbf{x}_j - \hat{\mathbf{x}}_j) = \mathbf{0}. \end{cases} \tag{11}$$

This system can be solved using a Newton procedure where $\mathbf{x}_j$ is initialized to $\hat{\mathbf{x}}_j$ and it is iteratively updated by the $\Delta\mathbf{x}_j$ increments satisfying

$$\begin{bmatrix} \mathbf{J}_j \\ \mathbf{\Phi}_i^\top \end{bmatrix} \Delta\mathbf{x}_j = - \begin{bmatrix} \mathbf{F}(\mathbf{x}_j) \\ \mathbf{\Phi}_i^\top \, (\mathbf{x}_j - \hat{\mathbf{x}}_j) \end{bmatrix} , \tag{12}$$

until the error is negligible or for a maximum number of iterations (Rheinboldt, 1996). The procedure is guaranteed to converge in a non-null, local neighborhood around $\mathbf{x}_i$.

The logarithmic map ψ_i^{-1} for a given point $\mathbf{x}_j$ in the neighborhood of $\mathbf{x}_i$ is given by the projection of $\mathbf{x}_j$ on the tangent space at $\mathbf{x}_i$

$$\mathbf{t}_j = \psi_i^{-1}(\mathbf{x}_j) = \mathbf{\Phi}_i^\top (\mathbf{x}_j - \mathbf{x}_i). \tag{13}$$

For a point $\mathbf{x}$ not in the manifold but close enough to $\mathbf{x}_i$, the mapping $\psi_i \circ \psi_i^{-1}$ produces a point on the manifold. Using this property, each step along a path $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ can be computed as

$$\mathbf{x}_j = \psi_i(\psi_i^{-1}(\mathbf{x}_{j-1} + \Delta\mathbf{v}_j)). \tag{14}$$

with $j > 1$, $\mathbf{x}_1$ given (initially $\mathbf{x}_s$), $\mathbf{x}_i$ the center of the chart used to define the exponential and logarithmic maps (initially $\mathbf{x}_s$ too), and $\Delta\mathbf{v}_j$ a variation of some of the actuated parameters. Typically, these parameters are the lengths (Tibert and Pellegrino, 2003b), the rest lengths (Sultan and Skelton, 2003), or the forces and torques applied to some of the elements of the tensegrity structure (Sultan, 2014). In any case, the steps should be small enough, i.e., $\|\Delta\mathbf{v}_j\| < \delta$ with a given parameter δ, so that the linear interpolation between configurations $\mathbf{x}_{j-1}$ and $\mathbf{x}_j$ is close to $\mathcal{E}$ and it can be safely assumed to be collision-free, provided that both configurations are non-colliding.

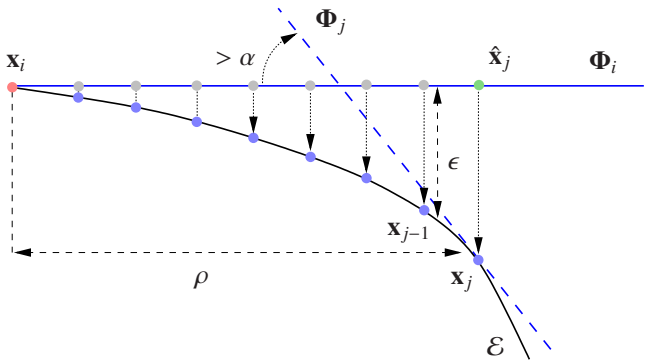


Figure 3: The tangent space at $\mathbf{x}_i$ can be used to define the exponential map as long as the distance and the curvature of $\mathcal{E}$ with respect to this tangent space are lower than ϵ and α, respectively. Moreover the maximum span is limited to ρ to obtain a regular paving of the manifold. In the figure, $\mathbf{x}_j$ does not fulfill these conditions and thus, a new tangent space will be defined at $\mathbf{x}_{j-1}$ and it will be used to continue the path on $\mathcal{E}$.

A step of size δ in control space may produce a larger step on $\mathcal{E}$. The tangent space at $\mathbf{x}_i$ can be safely used to define the exponential and logarithmic maps as long as this distortion is bounded, i.e, while

$$\| \hat{\mathbf{x}}_j - \mathbf{x}_j \| \leq \epsilon, \tag{15}$$

$$\| \mathbf{\Phi}_i^\top \mathbf{\Phi}_j \| \leq \cos(\alpha), \tag{16}$$

and

$$\| \mathbf{t}_j \| \leq \rho. \tag{17}$$

The first condition limits the distance to the tangent space, the second bounds the curvature error, and the last one is introduced to obtain a regular paving of the manifold. When Eqs. (15) to (17) do not hold, a new tangent space is defined at the last valid configuration and the path extension continues using the new chart. Figure 3 illustrates this process.

To keep track of the part of the manifold already covered by a given path, the charts created along the path can be coordinated. Following Henderson (2002), the area of the manifold parametrized by a given chart is represented by a polytope $\mathcal{P}_i$ defined in tangent space by a set L_i of linear inequalities. The set of inequalities for a chart is initially empty and is extended as new charts are defined around it. If a chart is created at a point $\mathbf{x}_j$ fulfilling Eqs. (15) to (17) with respect to a chart at $\mathbf{x}_i$, the inequality

$$2\, \mathbf{t}^\top \mathbf{t}_j \leq \|\mathbf{t}_j\|^2 \,, \tag{18}$$

with $\mathbf{t} \in \mathbb{R}^k$ is added to L_i, as shown in Fig. 4.

4.2. Defining an RRT on the equilibrium manifold

One can use an exhaustive exploration strategy to generate the full atlas of a given manifold. Since the atlas includes the neighboring relations between charts, once it is available, path planning queries can be readily solved using graph search algorithms. For instance, the symmetric tensegrity structure shown in Fig. 5-left has a two-dimensional equilibrium manifold, which can be described analytically (Sultan et al., 2001). The atlas of this manifold is shown in the right part of Fig. 5. In this

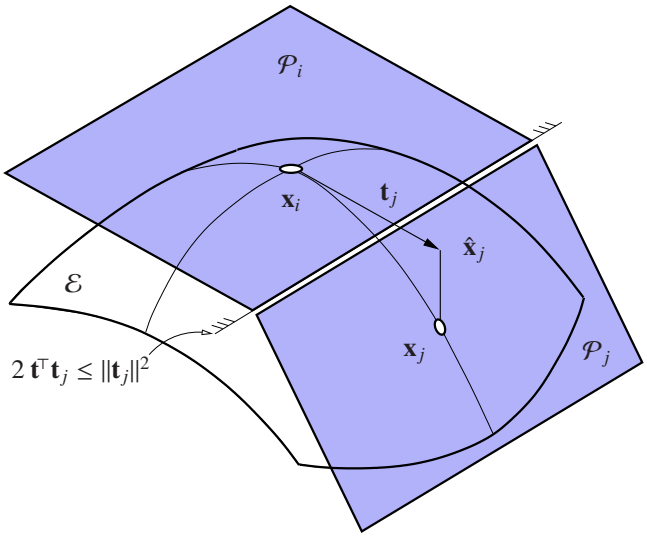


Figure 4: When a new chart is defined at a point $\mathbf{x}_j$ fulfilling Eqs. (15) to (17) with respect to a chart at $\mathbf{x}_i$, the polytopes associated with the two charts, $\mathcal{P}_i$ and $\mathcal{P}_j$, are coordinated to keep track of the area of $\mathcal{E}$ already explored. For instance, $2\,\mathbf{t}^\top\mathbf{t}_j \leq \|\mathbf{t}_j\|^2$ is added to L_i to crop $\mathcal{P}_i$. $\mathcal{P}_j$ is cropped in a similar way.

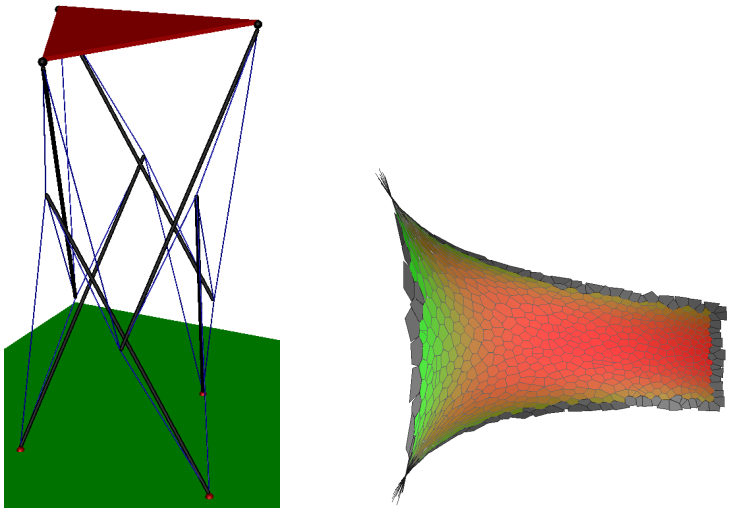


Figure 5: (Left) A two-stage symmetric tensegrity tower. (Right) Atlas of the equilibrium manifold of this structure. Each polygon is the applicability area $\mathcal{P}_i$ of the corresponding chart. The colors represent the compression force for the struts with green for low values and red for high ones. The gray charts are at the border of the ambient space, $\mathcal{A}$. This is equivalent to Fig. 6 in (Sultan et al., 2001). Once the atlas is computed, the neighboring relations between the charts define a graph which may be used to readily solve path planning queries.

case m is 94, n is 92 and the atlas is computed in less than 5 seconds in a standard desktop computer, but the execution time increases exponentially as the dimension of the manifold grows and the procedure soon becomes impractical.

The alternative proposed herein is to use a randomized strategy where the atlas construction is interleaved with the generation of an exploratory tree over $\mathcal{F}$. Thus,

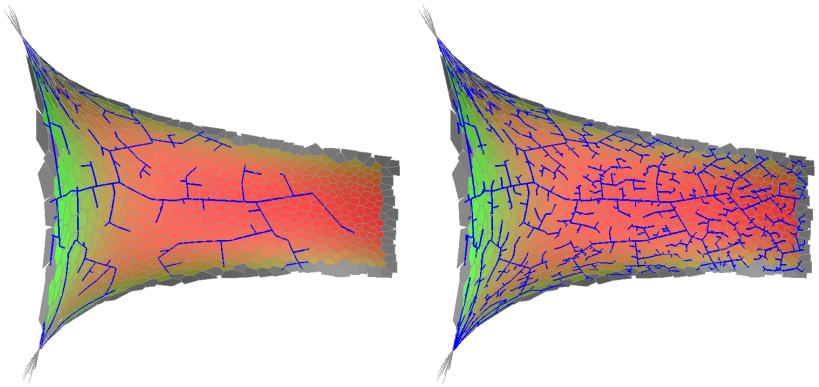


Figure 6: RRT on the equilibrium manifold of the symmetric tensegrity structure shown in Fig. 5-left. Left: An RRT with 1000 nodes. Right: An RRT with 3000 nodes.

only the part of the atlas necessary to connect the query configurations is actually built, which significantly accelerates the search. To this end we will adapt the RRT algorithm (LaValle and Kuffner, 2000). This algorithm initializes a tree $\mathcal{T}$ with $\mathbf{x}_s$ and repeats three steps: random sampling, nearest-neighbor search, and tree extension.

The sampling step requires to generate a random configuration $\mathbf{x}_r$ in the space to explore. However, in general, $\mathcal{E}$ can not be directly sampled since it is non-parametric. A rough approximation is to sample in $\mathcal{A}$ (Berenson et al., 2011), but this produces many ineffective tree extensions. A better alternative is to sample using the charts available up to a given moment. The tangent spaces provide a first-order approximation of $\mathcal{E}$ and, therefore, the samples generated using them are close to those that would be obtained if this manifold could be directly sampled. Thus, to obtain a random sample, one of these charts, say the i-th one, is selected at random and a vector of parameters $\mathbf{t}_r$ is sampled in a ball of radius $\rho_s > \rho$, rejecting the parameters that do not fulfill the linear inequalities defining the corresponding $\mathcal{P}_i$. If $\mathbf{t}_r$ is rejected, the whole procedure is repeated from scratch. Finally, $\mathbf{x}_r$ is computed as the ambient space coordinates of the accepted parameters, using Eq. (10).

The second step in the RRT construction is the identification of $\mathbf{x}_n$, the node already in $\mathcal{T}$ closer to $\mathbf{x}_r$. This should be done using the geodesic distance of $\mathcal{E}$. The implementation of an efficient nearest-neighbor procedure using geodesic distances is a challenging task (Chaudhry and Ivanov, 2010). A simpler and yet effective solution that will be adopted herein is to resort to the nearest-neighbor according to the Euclidean distance in $\mathcal{A}$, despite this may sometimes lead to unsuccessful tree extensions.

The last step in the RRT planner is to extend the tree advancing from $\mathbf{x}_n$ to $\mathbf{x}_r$. In parametric configuration spaces this is achieved by linear interpolation. In our case, we propose to use the procedure described in Section 4.1 to navigate over $\mathcal{E}$, taking steps of size δ in tangent space from $\mathbf{t}_n = \psi_i^{-1}(\mathbf{x}_n)$ to $\mathbf{t}_r = \psi_i^{-1}(\mathbf{x}_r)$, with i the index of the chart parameterizing $\mathbf{x}_n$. The branches are grown while they are on $\mathcal{E} \subset \mathcal{A}$ and until a collision is detected or the random sample is reached. If a branch is close enough to $\mathbf{x}_g$, a solution path can be readily determined traversing the tree up to the root.

Figure 6 shows the evolution of the RRT on the equilibrium manifold of the tensegrity structure shown in Fig. 5-left. As the number of samples increases, the underlying manifold gets more densely explored and the tree is more likely to include a path connecting the start and goal configurations. To increase the chance of reaching the goal several heuristics can be used. In the simplest one, the goal is used as a random sample

with a low but not null probability. A more effective strategy, adopted herein, is to grow two trees, one from $\mathbf{x}_s$ and one from $\mathbf{x}_g$. The trees are extended alternatively and after each extension a connection is attempted from the last node added to the tree to the nearest node in the other tree. The path planning problem is solved when the two trees are connected.

As all the randomized planning approaches, the AtlasRRT planner has difficulties in traversing the so called narrow corridors, i.e., small parts of $\mathcal{F}$ that must be necessarily traversed to connect the query configurations. Parameter ρ_s regulates the exploration behavior of the proposed planner: the larger its value the more aggressive the exploration, but the lower the capability of capture fine details like narrow corridors.

4.3. Path Optimization

The set S of stretchable elements in a tensegrity structure define the potential energy of a given configuration $\mathbf{x}_i$

$$E_i = \frac{1}{2} \sum_{j \in S} k_{i,j} \, (l_{i,j} - r_{i,j})^2, \tag{19}$$

and its gradient

$$\nabla E_i = (0, \ldots, 0, k_{i,j} \, (l_{i,j} - r_{i,j}), 0, \ldots, 0, -k_{i,j} \, (l_{i,j} - r_{i,j}), 0, \ldots, 0) \tag{20}$$

defines a force field. ∇E_i is a vector with non-null entries only at the positions corresponding to $l_{i,j}$ and $r_{i,j}$ for $j \in S$ and it can be interpreted as a drift since, it determines the uncontrolled underlying motion of the tensegrity structure (Ko et al., 2014). Thus, the control at configuration $\mathbf{x}_i$, $\mathbf{c}_i$, has to compensate ∇E_i while moving to the goal

$$\mathbf{c}_i = \mathbf{s}_i - \nabla E_i, \tag{21}$$

with $\mathbf{s}_i$ the unit speed vector tangent to the path at $\mathbf{x}_i$. In this way,

$$\mathbf{c}_i + \nabla E_i = \mathbf{s}_i, \tag{22}$$

i.e., the combined effect of the control and the drift is to follow the desired path.

The control effort for a trajectory $\pi : [0, 1] \rightarrow \mathcal{F}$ with $\pi(0) = \mathbf{x}_s$, $\pi(1) = \mathbf{x}_g$ can be defined as

$$C = \int_{t=0}^{1} \| \, \mathbf{c}_t \, \|^2 \, dt, \tag{23}$$

with $\mathbf{c}_t$ the control for configuration $\pi(t)$. This integral can be approximated on a path P with p discrete steps, $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ with $\mathbf{x}_1 = \mathbf{x}_s$ and $\mathbf{x}_p = \mathbf{x}_g$, assuming that the control is constant between two consecutive steps

$$C \approx \sum_{i=1}^{p-1} \mathbf{c}_i^\top \mathbf{c}_i \, d_i, \tag{24}$$

with $\mathbf{x}_i \in P$, $\mathbf{s}_i = (\mathbf{x}_{i+1} - \mathbf{x}_i)/d_i$, and $d_i = \|\mathbf{x}_{i+1} - \mathbf{x}_i\|$.

It can be shown that, if all the elements in the tensegrity structure are stretchable, which is the common situation in many formalizations, the equilibrium manifold is also a minimum energy manifold, i.e., $\nabla E = \mathbf{0}$ for all the valid configurations. Then, C simplifies to

$$C \approx \sum_{i=1}^{p-1} d_i, \tag{25}$$

which is the length of the path, i.e., the usual measure used to optimize paths in Robotics (Jaillet and Porta, 2013a).

Since the extremes of P are fixed, the gradient of C is

$$\nabla C = (\mathbf{0}, \ldots, \frac{\partial C}{\partial \mathbf{x}_i} \ldots, \mathbf{0}), \tag{26}$$

with

$$\frac{\partial C}{\partial \mathbf{x}_i} = 2\,\mathbf{c}_{i-1}^\top\,\mathbf{\Phi}_{i-1}^\top\,(\mathbf{I}_m - \mathbf{s}_{i-1}\,\mathbf{s}_{i-1}^\top) + \mathbf{c}_{i-1}^\top\,\mathbf{c}_{i-1}\,\mathbf{s}_{i-1}^\top - 2\,\mathbf{c}_i^\top\,\mathbf{\Phi}_i^\top\,(\mathbf{I}_m - \mathbf{s}_i\,\mathbf{s}_i^\top + \mathbf{H}_i\,d_i) - \mathbf{c}_i^\top\,\mathbf{c}_i\,\mathbf{s}_i^\top$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix and $\mathbf{H}_i$ is the Hessian of the potential energy function at $\mathbf{x}_i$.

An iterative gradient-descent method can be used to reduce the cost of a path updating it with small steps against ∇C. Formally, at a given iteration k each step of the path $P^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_p^k\}$ is updated as

$$\hat{\mathbf{x}}_i^{k+1} = \mathbf{x}_i^k - \lambda\,\frac{\partial C}{\partial \mathbf{x}_i} \tag{27}$$

for a small constant λ. Typically, $\hat{\mathbf{x}}_i^k$ is not on $\mathcal{E}$, but a new equilibrium configuration can be determined as

$$\mathbf{x}_i^{k+1} = \psi_i(\psi_i^{-1}(\hat{\mathbf{x}}_i^{k+1})), \tag{28}$$

defining a path P^{k+1} with lower control effort. The process is iterated while ∇C is non-null. Finally, note that to obtain a valid optimized path, points should not leave $\mathcal{F}$ and, thus, if $\mathbf{x}_i^k \in \mathcal{F}$ but $\mathbf{x}_i^{k+1} \notin \mathcal{F}$, $\mathbf{x}_i^{k+1}$ is set to $\mathbf{x}_i^k$ and $\frac{\partial C}{\partial \mathbf{x}_i}$ is set to $\mathbf{0}$ in the subsequent iterations of the gradient descent process.

5. Implementation

The path planning and path optimization methods for tensegrity structures proposed in this paper have been implemented in C and integrated in the CuikSuite (Porta et al., 2014), a software toolbox for position analysis and path planning under constraints in Robotics and Structural Biology. The source of the CuikSuite can be downloaded from (CuikSuite, web page) and it includes documentation to replicate the experiments in Section 6.

The CuikSuite has a high level format to describe tensegrity structures. For instance, Example 1 describes a basic kite-like structure in the CuikSuite formalism. This formalism is quite flexible and, besides the basic tensegrity elements, it permits defining information such as the external forces for each node, the set of anchored nodes, the set of nodes to be force-balanced, the symmetric elements, the global pretension (i.e., the norm of the vector of compressions and tensions), the obstacles, or the (self-)collisions to avoid.

The command

```
> cuikequations basic_kite.tens
```

reads the `basic_kite.tens` file and generates the set of equations described in Section 3. The software applies trivial simplifications to reduce the number of variables and constraints in the problem, whenever possible.

```
[constants]
 ls:=1      % length of struts
 fs:=-4     % max. compression
 lc:=2      % max. length of cables
 sc:=1      % stiffness of cables
[planar structure]
 strut s1:  n1 n2
            length ls
            force [fs,0.1*fs]
 strut s2:  n3 n4
            length ls
            force [fs,0.1*fs]
 cable c1:  n1 n3
            length [0.1*lc,lc]
            stiffness sc
            rest [0,lc]
 cable c2:  n2 n3
            length [0.1*lc,lc]
            stiffness sc
            rest [0,lc]
 cable c3:  n1 n4
            length [0.1*lc,lc]
            stiffness sc
            rest [0,lc]
 cable c4:  n2 n4
            length [0.1*lc,lc]
            stiffness sc
            rest [0,lc]
[fixed points]
 n1=(0,0)
 n2=(ls,0)
```

Example 1: The `basic_kite.tens` file describing the planar kite-like tensegrity structure in the CuikSuite formalism.

The command

```
> cuikexplore basic_kite.tens
```

implements the path tracing procedure described in Section 4.1 opening a graphical interface where the user can change the value of one of the actuated parameters at a time.

The CuikSuite also includes a MATLAB implementation of the form-finding procedure described in (Hernández-Juan and Mirats-Tur, 2008). This procedure is particularly well-suited for path planning purposes since, in contrast to other approaches, it can take into account (self-)collisions and it can fix the pose of some of the elements of the structure (e.g., the pose of the upper triangle in the structure shown in Fig. 5-left). Therefore, the method is able to generate configurations in $\mathcal{F}$ with particular properties, which is exactly what is necessary to determine $\mathbf{x}_s$ and $\mathbf{x}_g$. In this form-finding approach, the configuration of the tensegrity structure, $\mathbf{y}$, includes the coordinates of the nodes of the tensegrity framework, and the rest-length for each tensegrity element. This configuration is constrained by a set of equalities, H, to enforce the static equilibrium at each node and to fix the pose of selected tensegrity elements. A set of inequalities, G, are used to bound the lengths, the rest lengths, and the forces of the tensegrity elements and to set a minimum clearance between cylinders enclosing the struts and between the nodes and the planes representing the objects in the environment. Then,

the problem of determining a point fulfilling this set of constraints is transformed into an unconstrained optimization problem

$$\min_{\mathbf{y}} \; c \sum_{h \in H} \|h(\mathbf{y})\|^2 + \sum_{g \in G} \frac{1}{c} \left(e^{-c\, g(\mathbf{y})} - 1\right) + E_{\mathbf{y}} \tag{29}$$

using quadratic and exponential functions for the equality and inequality constraints, respectively, all of them weighted by a penalty factor, c, and where $E_{\mathbf{y}}$ is the elastic energy of configuration $\mathbf{y}$ as defined in Eq. (19). This unconstrained problem is iteratively solved starting from a random configuration and applying a quasi-Newton method where the penalty factor is increased at each iteration until a solution is found. Thus, the constraints are relaxed at the beginning and they play a more relevant role as the optimization advances.This procedure is guaranteed to converge, at least, to a local minimum.

Once the start and goal configurations are fixed, the command

```
> cuikatlasrrt basic_kite.tens
```

solves the path planning problem with the method described in Section 4.2. The output of this command is a file `basic_kite_path.sol` with the points in $\mathcal{E}$ defining the path from the start to the goal configurations. Although methods to detect self-collisions in tensegrity structures exist (Cefalo and Mirats-Tur, 2010), in our case we also need to avoid collisions with objects in the environment. Thus, during the planing the possible presence of collisions is checked with highly-efficient, general collision detection libraries (Pan et al., 2012).

Finally, the command

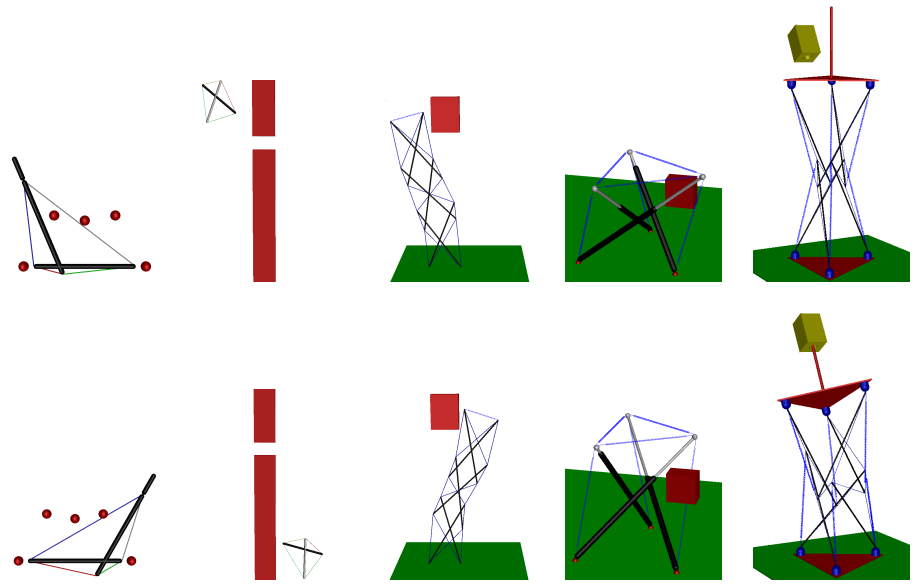
```
> cuiksmoothpath basic_kite.tens basic_kite_path.sol effort
```

optimizes the path with the method described in Section 4.3.

The CuikSuite includes many additional tools to manipulate and visualize the solution paths. For more details see (Porta et al., 2014) and the on-line documentation (CuikSuite, web page).

6. Experiments

Figure 7 shows the five tensegrity structures used to test the planner introduced in this paper. The first test case is a planar kite-like structure, where one of the struts is fixed horizontally and the other must not collide with the red spheres in the environment. This is the simplest possible tensegrity structure and has a 3-dimensional equilibrium manifold where paths can be directly visualized. The second problem is a kite where the nodes are not fixed and, thus, it can freely translated and rotate in the plane. The objective is to traverse a narrow corridor where the structure has to be almost folded and, thus, this is a particularly hard case for the probabilistic planners, such as the one proposed herein. The third benchmark is a planar tensegrity tower fixed to the ground also used by Wijdeven and Jagerm (2005). In the setting used here, the structure has to fold and unfold to move around an obstacle. No symmetry is considered and, thus, the equilibrium manifold is of high dimension. Therefore, this example is used to illustrate the scalability of the approach with respect to the test-cases used in previous path planners for tensegrity structures (Hernández-Juan and Mirats-Tur, 2008; Xu et al., 2014). As a reference, none of these existing approaches have been proven to be effective in equilibrium manifolds of dimension higher than 3. The fourth



(a) Kite (b) Free-flying (c) Tower (d) Prism (e) Robot

Figure 7: The five benchmarks used in this paper. (a) A planar kite-like structure. (b) A planar free-flying structure. (c) A planar tower. (d) A 3D prism. (e) A six degree of freedom robot. The top and bottom rows show the start and goal configurations, respectively

example is a basic three-dimensional tensegrity structure with three nodes fixed to the ground that has been used in many active structures (Hagiwara and Oda, 2010; Mirats-Tur and Camps-Sala, 2011). The mobility of this structure depends on the actual set of active elements. In the particular example shown here, the structure is actuated by three prismatic struts and the objective is to avoid self-collisions and collisions between the structure's elements and the red box in the environment. Finally, the fifth example is a manipulator with fixed base and with six degrees of freedom, modeled after the robot described in (Tensegriteam, web page). The structure is composed by two three-dimensional prisms properly connected to ensure stability and it has to insert a peg into a hole. A similar structure has been used in flight simulators (Sultan et al., 2000).

The benchmarks have ambient spaces, sets of equations, and equilibrium manifolds of different dimensions, as well as different obstacles settings, with the aim of proving the generality of the proposed planning approach. To prove the robustness of the method, the same set of parameters are used in all the experiments: $\epsilon = 0.1$, $\cos(\alpha) = 0.1$, $\rho = 0.4$, $\delta = 0.05$, and $\rho_s = k$.

Table 1 summarizes the basic information about the test cases such as the number of nodes and the number of fixed-length and stretchable elements in the abstract tensegrity framework, the number of variables and constraints in the corresponding system of equations after simplifying it, and the dimension of the equilibrium manifold. The table also shows the number of charts and nodes generated by the AtlasRRT planner and the planning time in seconds for the five benchmarks on a MacBook Pro laptop with an i7 processor at 2.66 Ghz. The execution time of probabilistic planners typically varies significantly between different runs. Therefore, the results in the table correspond to the average over 10 runs. The path planning process was successful in all the repetitions, which proves the reliability of the proposed planner.

Benchmark	q_n	q_f	q_s	m	n	k	charts	nodes	t_p	C	C^*	t_o
(a) Kite	4	2	4	26	23	3	165	1456	0.16	18.21	11.37	10
(b) Free-flying	4	2	4	28	22	6	3368	34053	25.59	50.68	20.19	28
(c) Tower	12	6	18	106	106	15	229	2620	8.69	25.74	9.54	120
(d) Prism	6	3	6	32	30	2	17	167	0.04	4.65	2.89	5
(e) Robot	12	6	24	144	138	6	175	1576	5.08	11.12	4.81	85

Table 1: For the five benchmarks used in this paper: The number of nodes (q_n), of elements with fixed length (q_f), and of stretchable elements (q_s), the dimension of the ambient space (m) and the number of equations (n) after simplifying the problem, the dimension of the equilibrium manifold (k), the number of charts, the number of RRT nodes, the average planning time in seconds (t_p), the control effort of a typical planned path (C), the control effort of the optimized path (C^*), and, finally, the optimization time in seconds (t_o).

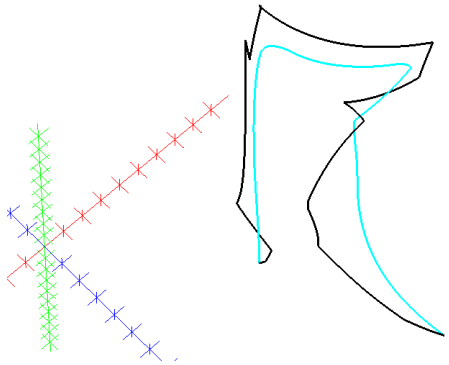


Figure 8: Planned path (in black) and optimized path (in cyan) for the Kite example.

As a reference, the RRT path planner described by Xu et al. (2014) takes more than 800 seconds to solve a trivial path planning problem with a kite structure, while the approach described here takes less than 0.05 seconds to solve the same problem (CuikSuite, web page). Although the computers and the programming languages used are not the same in the two approaches, the main difference between them can be attributed to the use of a computationally complex relaxation procedure in (Xu et al., 2014), while the method introduced herein to explore the equilibrium manifold is comparatively inexpensive.

The planner is effective exploring the equilibrium manifold, but it has difficulties in traversing narrow corridors. For this reason, the planning time of the Free-flying system is higher than the planning time of the Tower example, despite the former has an ambient space of lower dimension, a smaller set of equations, and a lower-dimensional equilibrium manifold. Thus, the determining factor for the planning time is the complexity of the environment rather than the characteristics of the equilibrium manifold. The complexity of the obstacle setting can be tackled by adjusting parameter ρ_s either experimentally or using adaptive sampling techniques (Porta and Jaillet, 2014).

Table 1 also includes the control effort for a typical path directly returned by the planner (C), the effort for this path optimized with the method introduced in Section 4.3 (C^*) as well as the time in seconds taken by the optimization process (t_o). This time is relatively large compared with the planning time because the paths are densely sampled, the gradient must be computed for all the points in the path, and the gradient descent steps are small. As a reference, a typical output path for the Tower benchmark includes about 600 steps. The cost of the optimization process could be

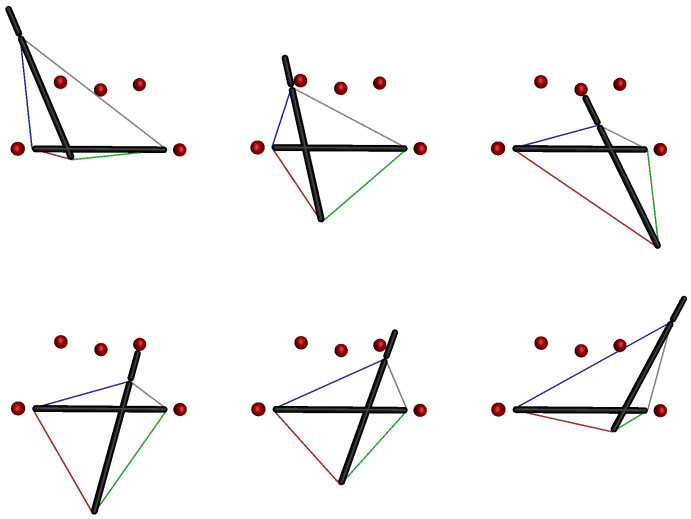


Figure 9: Snapshots of the execution of the optimized path for the Kite example.

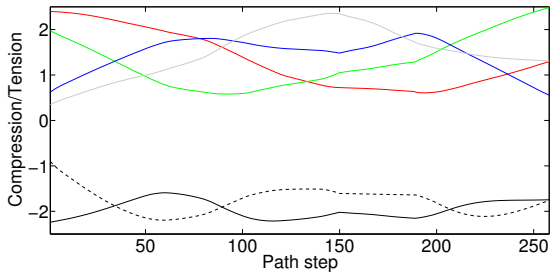


Figure 10: Compression (negative) and tension (positive) forces for the struts and cables of the Kite benchmark. For the compressed elements, the solid line correspond to the horizontal strut and the dashed line to the strut with variable orientation. For the cables, the color code of Fig. 7(a) is used.

reduced sub-sampling the paths, but then the approximation of the path effort will be coarser.

Figure 8 shows a typical solution path obtained with the planner for the Kite example in black and the finally optimized path in cyan. The paths are plotted in the subspace of the lengths of the cables shown in red, green, and blue in Fig 7(a). Clearly, the initial path is long and includes sharp direction changes, while the optimized path is shorter and smoother and, thus, suitable for execution on the real structure. Figure 9 shows different configurations along this optimized path. Finally, Fig. 10 shows the evolution of the compression and tension forces along this path. As expected, the forces evolve smoothly, they are negative for compressions and positive for tensions, and they are always inside the prescribed ranges for the corresponding variables. Thus, the structure is properly stressed at any time. Similar results can be obtained for the rest of benchmarks.

7. Conclusions

This paper deals with the problem of path planning for tensegrity structures, a fundamental issue hardly addressed in the literature so far. The presented results show that

the planner described herein is general, efficient, and scales to problems significantly more complex than those solved by previous approaches. As in all the probabilistic planning approaches, the main bottleneck for this planner arises from the arrangement of obstacles in the environment. Thus, the planner effectively overcomes the problem of dealing with the equilibrium manifold, which has been often pointed as the major hurdle for the use of probabilistic planners in the context of tensegrity structures (Komendera, 2013). Besides the planning method, this paper also introduced a path optimization technique. The optimized path can be directly fed to existing control algorithms and, thus, the proposed approach nicely complements such algorithms.

An aspect not considered in this paper is the possible presence of singularities, i.e., configurations where the number of mechanisms, i.e., self-motions, of the tensegrity structure changes (Bohigas et al., 2014). In such configurations the kinetostatic performance of the framework dramatically degrades causing control issues. In extreme situations, even the integrity of the structure can be in risk. Two strategies are possible to deal with singularities. Either special planning and control procedures are used to traverse them (Porta et al., 2012; Jui and Sun, 2005) or they are avoided (Bohigas et al., 2013). Both strategies can be based on higher-dimensional continuation and it is our future endeavor to integrate them in the planner for tensegrity structures introduced herein.

Finally, when the tensegrity structure has to move at high speed or has to interact with objects in the environment or with the ground, dynamic constraints have to be considered. Such constraints define a manifold in a space involving not only positions and forces, but also speeds, and accelerations (LaValle, 2006). We plan to investigate the possible use of the higher-dimensional continuation tools to explore this manifold.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under project DPI2014-57220-C2-2-P. We would like to thank Lluís Ros for his valuable comments during the elaboration of this paper.

References

Berenson, D., Srinivasa, S. S., Kuffner, J. J., 2011. Task space regions: A framework for pose-constrained manipulation planning. International Journal of Robotics Research 30 (12), 1435–1460.

Bohigas, O., Henderson, M. E., Ros, L., Manubens, M., Porta, J. M., 2013. Planning singularity-free paths on closed-chain manipulators. IEEE Transactions on Robotics 29 (4), 888–898.

Bohigas, O., Zlatanov, D., Ros, L., Manubens, M., Porta, J. M., 2014. A general method for the numerical computation of manipulator singularity sets. IEEE Transactions on Robotics 30 (2), 340–351.

Bruce, J., Caluwaerts, K., Iscen, A., Sabelhaus, A. P., SunSpiral, V., 2014. Design and evolution of a modular tensegrity robot platform. In: IEEE International Conference on Robotics and Automation. pp. 3483–3489.

Cefalo, M., Mirats-Tur, J. M., 2010. Real-time self collision detection algorithms for tensegrity systems. International Journal of Solids and Structures 47 (13), 1711–1722.

Chaudhry, R., Ivanov, Y., 2010. Fast approximate nearest neighbor methods for non-Euclidean manifolds with applications to human activity analysis in videos. In: European Conference on Computer Vision. pp. 735–748.

Connelly, R., 1982. Rigidity and energy. Inventiones mathematicae 66 (1), 11–33.

Connelly, R., 2013. Tensegrities and global rigidity. Springer, Ch. 21, pp. 267–278.

Connelly, R., Terrell, M., 1995. Globally rigid symmetric tensegrities. Structural Topology 21, 59–78.

Connelly, R., Whiteley, W., 1992. Second-order rigidity and prestress stability for tensegrity frameworks. SIAM Journal on Discrete Mathematics 9 (3), 453–491.

CuikSuite, web page. http://www.iri.upc.edu/cuik.

do Carmo, M. P., 1976. Differential geometry of curves and surfaces. Prentice-Hall.

Fuller, R. B., 1962. Tensile-integrity structures. US Pattent no. 3,063,521.

Geraerts, R., Overmars, M. H., 2007. Creating high-quality paths for motion planning. The International Journal of Robotics Research 26, 845–863.

Gómez-Estrada, G., Bungartz, H.-J., Mohrdieck, C., 2006. Numerical form-finding of tensegrity structures. International Journal of Solids and Structures 43 (22-23), 6855–6868.

Hagiwara, Y., Oda, M., 2010. Deformation of a tensegrity structure using tendon actuators. In: International Symposium on Artificial Intelligence, Robotics and Automation in Space. pp. 699–706.

Han, L., Amato, N. M., 2000. A Kinematics-based probabilistic roadmap method for closed chain systems. In: Algorithmic and Computational Robotics - New Directions (WAFR2000). pp. 233–246.

Han, L., Rudolph, L., 2006. Inverse kinematics for a serial chain with joints under distance constraints. In: Robotics: Science and Systems. pp. 177–184.

Hanaor, A., 1992. Aspects of design of double layer tensegrity domes. International Journal of Space Structures 7 (2), 101–113.

Henderson, M. E., 2002. Multiple parameter continuation: Computing implicitly defined k-manifolds. International Journal of Bifurcation and Chaos 12 (3), 451–476.

Hernández-Juan, S., Mirats-Tur, J. M., 2008. A method to generate stable, collision free configurations for tensegrity based robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3769–3774.

Hernández-Juan, S., Skelton, R. E., Mirats-Tur, J. M., 2009. Dynamically stable collision avoidance for tensegrity based robots. In: ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots. pp. 315–322.

Ingber, D., 1993. Cellular tensegrity: defining new rules for biological design that govern the cytoskeleton. Journal of Cell Science 104, 613–627.

Jaillet, L., Porta, J., 2011. Path Planning with Loop Closure Constraints using an Atlas-based RRT. In: International Symposium on Robotics Research.

Jaillet, L., Porta, J. M., 2013a. Efficient asymptotically-optimal path planning on manifolds. Robotics and Autonomous Systems 61 (8), 797–807.

Jaillet, L., Porta, J. M., 2013b. Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Transactions on Robotics 29 (1), 105–117.

Jui, C. K. K., Sun, Q., 2005. Path tracking of parallel manipulators in the presence of force singularity. ASME Journal of Dynamic Systems, Measurement, and Control 127, 550–563.

Karaman, S., Frazzoli, E., 2011. Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research 30 (7), 846–894.

Ko, I., Kim, B., Park, F. C., 2014. Randomized path planning on vector fields. International Journal of Robotics Research 33 (13), 1664–1682.

Komendera, E., 2013. A Survey of the Computational Modeling and Control of Tensegrity Robots.

Koohestani, K., 2012. Form-finding of tensegrity structures via genetic algorithm. International Journal of Solids and Structures 49 (5), 739–747.

Koohestani, K., Guest, S. D., 2013. A new approach to the analytical and numerical form-finding of tensegrity structures. International Journal of Solids and Structures 50 (19), 2995–3007.

Korkmaz, S., Bel Hadj Ali, N., Smith, I. F. C., 2010. Self-repair of a tensegrity pedestrian bridge through grouped actuation. In: International Conference in computing and Building Engineering. pp. 3769–3774.

LaValle, S. M., 2006. Planning algorithms. Cambridge University Press, New York.

LaValle, S. M., Kuffner, J. J., 2000. Rapidly-exploring random trees: Progress and prospects. In: Algorithmic and Computational Robotics - New Directions (WAFR2000). pp. 293–308.

Liedl, T., Högberg, B., Tytell, J., Ingber, D., Shih, W., 2010. Self-assembly of three-dimensional prestressed tensegrity structures from DNA. Nature Nanotechnology 5, 520–524.

Micheletti, A., Williams, W., 2007. A marching procedure for form- finding for tensegrity structures. Journal of Mechanics of Materials and Structures 2 (5), 857–882.

Mirats-Tur, J. M., Camps-Sala, J., 2011. A 3-DOF actuated robot: A minimal tensegrity configuration. IEEE Robotics and Automation Magazine 18 (3), 96–113.

Moored, K. W., Bart-Smith, H., 2006. The analysis of tensegrity structures for the design of a morphing wing. Journal of Applied Mechanics 74 (4), 668–676.

Moored, K. W., Kemp, T. H., Houle, N. E., Bart-Smith, H., 2010. Analytical predictions, optimization, and design of a tensegrity-based artificial pectoral fin. International Journal of Solids and Structures 48 (22-23), 3142–3159.

Motro, R., 1984. Forms and forces in tensegrity systems. In: International Conference on Space Structures. pp. 180–185.

Motro, R., 2003. Tensegrity: Structural systems for the future. Butterworth-Heinemann.

Murakami, H., Nishimura, Y., 2001. Static and dynamic characterization of regular truncated icosahedral and dodecahedral tensegrity modules. International Journal of Solids and Structures 38 (50-51), 9359–9381.

Pagitz, M., Mirats-Tur, J. M., 2009. Finite element based form-finding algorithm for tensegrity structures. International Journal of Solids and Structures 46 (15), 3235–3240.

Pan, J., Chitta, S., Manocha, D., 2012. FCL: A general purpose library for collision and proximity queries. In: IEEE International Conference on Robotics and Automation. pp. 3859–3866.

Paul, C., Valero-Cuevas, F., Lipson, H., 2006. Design and control of tensegrity robots for locomotion. IEEE Transactions on Robotics 22 (5), 944–957.

Pinaud, J. P., Masic, M., Skelton, R. E., 2003. Path planning for the deployment of tensegrity structures. In: International Symposium on Smart Structures and Materials: Modeling, Signal Processing, and Control. p. 436.

Porta, J. M., Jaillet, L., 2014. Sampling strategies for path planning under kinematic constraints. arXiv:1407.2544.

Porta, J. M., Jaillet, L., Bohigas, O., 2012. Randomized Path Planning on Manifolds based on Higher-Dimensional Continuation. International Journal of Robotics Research 31 (2), 201–215.

Porta, J. M., Ros, L., Bohigas, O., Manubens, M., Rosales, C., Jaillet, L., 2014. The Cuik Suite: Analyzing the motion closed-chain multibody systems. IEEE Robotics and Automation Magazine 21 (3), 105–114.

Porta, J. M., Ros, L., Schulze, B., Sljoka, A., Whiteley, W., 2013. On the symmetric molecular conjectures. In: Workshop on Computational Kinematics. pp. 175–184.

Rheinboldt, W. C., 1996. MANPACK: A set of algorithms of computations on implicitly defined manifolds. Computers and Mathematics with Applications 32 (12), 15–28.

Rhode-Barbarigos, L., Schulin, C., Ali, N., Motro, R., Smith, I., 2012. Mechanism-based approach for the deployment of a tensegrity-ring module. Journal of Structural Engineering 138 (4), 539–548.

Samili, A., Motro, R., 2007. Foldable/Unfoldable Curved Tensegrity Systems by Finite Mechanism Activation. Journal of the International Association for Shell and Spatial Structures 48 (3), 153–160.

Skelton, R. E., Helton, J., Adhikari, R., Pinaud, J., Chan, W., 2001. An introduction to the mechanics of tensegrity structures. CRC Press, Ch. 17, pp. 316–388.

Snelson, K., 1965. Continuous tension, discontinuous compression structures. US Patent no. 3,169,611.

Snelson, K., web page. http://kennethsnelson.net.

Sultan, C., 2014. Tensegrity deployment using infinitesimal mechanisms. International Journal of Solids and Structures 51 (21-22), 3653–3668.

Sultan, C., Corless, M., Skelton, R. E., 1999. Peak to peak control of an adaptive tensegrity space telescope. In: Symposium on Smart Structures and Materials. pp. 190–201.

Sultan, C., Corless, M., Skelton, R. E., 2000. Tensegrity flight simulator. Journal of Guidance, Control, and Dynamics 26 (6), 1055–1064.

Sultan, C., Corless, M., Skelton, R. E., 2001. The prestressability problem of tensegrity structures: some analytical solutions. International Journal of Solids and Structures 38 (30-31), 5223–5252.

Sultan, C., Skelton, R. E., 2003. Deployment of tensegrity structures. International Journal of Solids and Structures 40 (18), 4637–4657.

Sultan, C., Skelton, R. E., 2004. A force and torque tensegrity sensor. Sensors and Actuators 11 (2), 220–231.

Tensegriteam, web page. https://seniordesign.engr.uidaho.edu/2010-2011/tensegriteam.

Tibert, A. G., Pellegrino, S., 2003a. Review of form-finding methods for tensegrity structures. International Journal of Space Structures 28 (4), 209–223.

Tibert, A. G., Pellegrino, S., 2003b. Deployable tensegrity masts. In: AIAA Structures, Structural Dynamics, and Materials Conference. pp. 1–10.

Tibert, G., 2003. Deployable tensegrity structures for space applications. Ph.D. thesis, Royal Institute of Technology.

Wijdeven, J. v. d., Jagerm, B. d., 2005. Shape change of tensegrity structures: Design and control. In: American Control Conference. pp. 2522–2527.

Williamson, D., Skelton, R. E., Han, J., 2003. Equilibrium conditions of a tensegrity structure. International Journal of Solids and Structures 40 (23), 6347–6367.

Wroldsen, A. S., de Oliveira, M. C., Skelton, R. E., 2009. Modelling and control of non-minimal non-linear realisations of tensegrity systems. International Journal of Control 82 (3), 389–407.

Xu, S., Sun, F., Luo, Y., Xu, Y., 2014. Collision-free path planning of tensegrity structures. Journal of Structural Engineering 140 (4).

Zhang, J. Y., Guest, S. D., Ohsaki, M., 2009a. Symmetric prismatic tensegrity structures. Part II: Symmetry-adapted formulations. International Journal of Solids and Structures 46 (1), 15–30.

Zhang, J. Y., Guest, S. D., Ohsaki, M., 2009b. Symmetric prismatic tensegrity structures: Part I: Configuration and Stability. International Journal of Solids and Structures 46 (1), 1–14.