Motion planning for 6-D manipulation with aerial towed-cable systems

Montserrat Manubens*, Didier Devaurs†‡, Lluís Ros* and Juan Cortés†‡
* Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
Email: {mmanuben, llros}@iri.upc.edu
† CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
‡ Univ de Toulouse, LAAS, F-31400 Toulouse, France
Email: {ddevaurs, jcortes}@laas.fr

***Abstract*—Performing aerial 6-dimensional manipulation using flying robots is a challenging problem, to which only little work has been devoted. This paper proposes a motion planning approach for the reliable 6-dimensional quasi-static manipulation with an aerial towed-cable system. The novelty of this approach lies in the use of a cost-based motion-planning algorithm together with some results deriving from the static analysis of cable-driven manipulators. Based on the so-called wrench-feasibility constraints applied to the cable tensions, as well as thrust constraints applied to the flying robots, we formally characterize the set of feasible configurations of the system. Besides, the expression of these constraints leads to a criterion to evaluate the quality of a configuration. This allows us to define a cost function over the configuration space, which we exploit to compute good-quality paths using the T-RRT algorithm. As part of our approach, we also propose an aerial towed-cable system that we name the *FlyCrane*. It consists of a platform attached to three flying robots using six fixed-length cables. We validate the proposed approach on two simulated 6-D quasi-static manipulation problems involving such a system, and show the benefit of taking the cost function into account for such motion planning tasks.**

I. Introduction

Aerial towed-cable systems have been used for decades, mainly as crane devices. They have proved to be very useful in various contexts, such as supply delivery missions and rescue operations [3], as well as environmental monitoring and surveillance [18]. One such system has even been successful as a safe soft-landing device for a rover on the martian surface [17], for instance. In all these examples, the systems only required a certain position accuracy, for example to execute simple trajectories [15, 13]. Little work has been done on trying to govern a load in both position and orientation. To the best of our knowledge, the only existing technique for 6-dimensional manipulation with an aerial towed-cable system requires a given discrete set of load poses [12, 7]. Such a technique relies on solving the inverse kinematics problem and determining the static equilibrium for all given poses. Requiring a given set of platform poses may be too restrictive, though, especially in constrained workspaces, because it may provide no result, while there may exist solutions for other intermediate poses.

This paper presents a new reliable motion planning approach for 6-dimensional quasi-static manipulation with aerial towed-cable systems. The method only requires a start and goal configurations as input, and provides a feasible path to achieve the manipulation task. In addition to being feasible, the generated manipulation path will be of *good quality*, meaning that all intermediate configurations fulfill adequate physical properties related to the forces applied to the system and to the cable tensions. This quality will be measured by a formal criterion derived from the static analysis of the system, based on a similar formulation as that used for cable-driven manipulators [6, 4]. A path-planing algorithm taking this quality measure into account [9] will then be applied to compute good-quality paths.

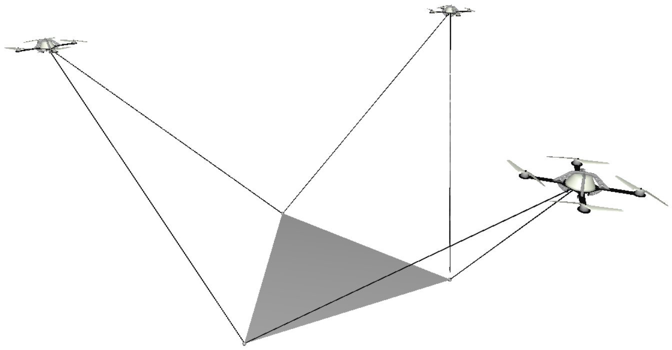

Fig. 1. Octahedral version of the *FlyCrane* system.

In addition to the methodology, this paper presents an aerial towed-cable system to perform 6-D manipulation tasks, that we call the *FlyCrane*. This system consists of a moving platform attached to three flying robots by means of six fixed-length cables linked by pairs to each robot. The 6-D manipulation of the platform can be performed by varying the relative positions of the flying robots. An octahedral version of this system is illustrated in Figure 1.

The rest of the paper is organized as follows: Section II provides an overview of our contribution, whose elements are detailed in Sections III and IV. Section V presents an evaluation of our approach on two 6-D manipulation problems involving the octahedral version of the *FlyCrane* system.

II. Overview of the contribution

Towed-cable systems present important analogies with cable-driven manipulators, which enable us to perform their static analysis in a way similar to that presented in [4]. However, while cable-driven manipulators have to adjust the lengths of their cables to reach a precise pose of the platform, towed-cable systems have fixed-length cables and are actuated by displacing their anchor points. Manipulating the six degrees of freedom of a load requires a minimum of seven cables, unless some convenient forces reduce this number. In crane configurations, for instance, gravity acts as an implicit cable, and therefore six cables suffice for the full 6-D manipulation. Examples of such structures are the NIST Robocrane [1] or more general cable-driven hexapods [4].

In the proposed aerial towed-cable system, called the *FlyCrane*, the platform is also pulled by six cables, which, as illustrated in Fig. 2, are pairwise attached to three flying robots (instead of attaching them individually to six flying robots). It is worth noting that three is the minimal number of flying robots required to properly operate this system, as less robots would not allow the manipulation of the six degrees of freedom of the platform. Whenever the cable base points are also coupled ($B_1 = B_2$, $B_3 = B_4$, $B_5 = B_6$), we call it *octahedral FlyCrane*, because the structure can be seen as an octahedron, comprising the following 8 triangles: the platform base points, the triangle formed by the flying robots, and the 6 triangles made of pairs of adjacent cables. Section III-A formalizes the notations describing the *FlyCrane*.

In this paper we assume that motions are performed quasi-statically, thus neglecting the dynamic analysis of the system. Although it may appear as a strong simplification, this assumption is frequently made in fine-positioning situations, where slow motion is imperative. Nevertheless, dealing with dynamical aspects can be an interesting extension for future work, as will be discussed in Section VI.

Even with six cables, the six degrees of freedom of the platform can be governed only in a subset of the configuration space of the system. Indeed, the pose of the platform is locally determined only when all cables are in tension. Therefore, it is important to prevent the cables from being slack or too tight. Besides, the flying robots must be able to counteract the forces exerted on them. These two conditions determine the feasibility of a configuration of the system. More precisely, to be feasible, a configuration must satisfy the following two types of constraints, that will be formalized in Section III-B:

- *Wrench-feasibility* constraints: they guarantee that the system is able to statically counteract a set of wrenches applied on the platform while ensuring that the cable tensions always lie within a pre-defined, positive acceptance range; they are derived from the static analysis of cable-driven manipulators [6, 4].
- *Thrust* constraints: they guarantee that the thrust of the flying robots can equilibrate the forces applied on them, namely the forces exerted by the cables and the force of gravity.

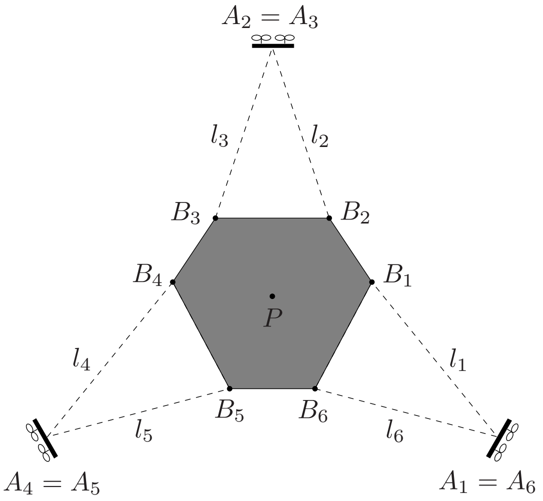

Fig. 2. Geometric structure of a generic *FlyCrane* system.

An infinite number of feasible solution paths may exist for a given manipulation query on such a system. A way to discriminate the less appropriate ones, is to define a criterion assessing their quality. A good-quality path should be a path whose intermediate configurations are attributed a low cost with respect to the physical properties of the system. A meaningful way to evaluate the cost of a configuration of the system is to derive it from the previous feasibility constraints, as will be explained in Section IV-A.

Any general path planner, such as the Rapidly-exploring Random Tree (RRT) algorithm [11], could be applied to compute collision-free paths satisfying the previous feasibility constraints to perform 6-D manipulation tasks with the *FlyCrane* system. However, it might not produce good-quality paths. Since we will define a cost function over the configuration space, we can use a cost-based path planner, such as the Transition-based RRT (T-RRT) [9], in order to obtain good-quality manipulation paths. T-RRT has been successfully applied to various types of problems in robotics [9, 2] and structural biology [10]. But, it is worth noting that, to the best of our knowledge, this is the first time it is applied to aerial manipulation problems.

III. System description and constraints

This section presents the *FlyCrane* system. First, we introduce some notations and provide a description of the system. Then, we formulate the constraints ensuring the feasibility along the motion paths.

A. Description of the system

The *FlyCrane* system consists of a platform attached to six cables of fixed lengths l_i. Each cable is attached to the platform and to a flying robot at points B_i and A_i, respectively. Each flying robot is tied to two cables so that $A_1 = A_6$, $A_2 = A_3$ and $A_4 = A_5$, as can be seen in Fig. 2.

Let $OXYZ$ and $PX'Y'Z'$ be the fixed and the moving reference frames attached to the ground and to the platform,

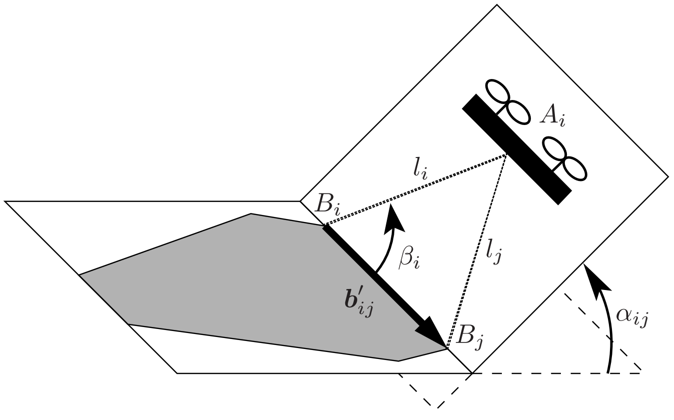

Fig. 3. α_{ij} measures the angle between the plane of the platform and the plane of cables i and j attached to the same flying robot.

respectively. Any configuration of the system can be uniquely represented by $\boldsymbol{q} = (\boldsymbol{p}, \boldsymbol{R}, \boldsymbol{\alpha}) \in \mathcal{C} = SE(3) \times \mathbb{T}^3$, where $\boldsymbol{p} = [x, y, z]^\mathsf{T}$ is the position vector of a point P on the platform expressed in the fixed frame, $\boldsymbol{R}$ is a 3×3 rotation matrix providing the orientation of $PX'Y'Z'$ relative to $OXYZ$, and $\boldsymbol{\alpha} = [\alpha_{16}, \alpha_{23}, \alpha_{45}]^\mathsf{T}$, where each α_{ij} is the dihedral angle between the plane formed by cables i and j and the plane of the platform (see Fig. 3). The entries of $\boldsymbol{R}$ can be defined in a variety of ways. Due to its advantageous properties [5], here we use the parameterization provided by tilt-and-torsion angles, $\boldsymbol{\tau} = \{\phi, \theta, \sigma\}$, for which

$$\boldsymbol{R} = \boldsymbol{R}_z(\phi)\, \boldsymbol{R}_y(\theta)\, \boldsymbol{R}_z(\sigma - \phi). \tag{1}$$

Let A_i and B_i be the points where the ith cable is attached to the flying robot and to the platform respectively (Fig. 2). We denote by $\boldsymbol{a}_i$ and $\boldsymbol{a}'_i$ the position vectors of point A_i expressed in $OXYZ$ and in $PX'Y'Z'$, respectively. Similarly, $\boldsymbol{b}_i$ and $\boldsymbol{b}'_i$ will denote the position vectors of point B_i expressed in $OXYZ$ and in $PX'Y'Z'$, respectively. Then, the coordinates of points A_i can be expressed as

$$\boldsymbol{a}_i = \boldsymbol{p} + \boldsymbol{R}\, \boldsymbol{a}'_i, \tag{2}$$

where $\boldsymbol{a}'_i = \boldsymbol{b}'_i + l_i\, \boldsymbol{R}_{\boldsymbol{b}'_{ij}}(\alpha_{ij})\, \boldsymbol{R}\boldsymbol{n}(\beta_i)\, \frac{\boldsymbol{b}'_{ij}}{\|\boldsymbol{b}'_{ij}\|}$, with $\boldsymbol{b}'_{ij}$ being the vector $B_j - B_i$ expressed in $PX'Y'Z'$, $\boldsymbol{n}$ being the unit vector normal to the platform, and β_i being the angle of cable i with respect to vector $\boldsymbol{b}'_{ij}$.

Finally, we have to consider that some external forces $\mathbf{f}$, as gravity for instance, are applied on the load of the platform. Any of these forces and its associated torque constitute what is called a *wrench*, $\hat{\boldsymbol{w}} = [\mathbf{f}^\mathsf{T}, (\mathbf{f} \times \boldsymbol{p})^\mathsf{T}]^\mathsf{T}$. We denote by $\hat{\boldsymbol{w}}_0$ the resultant applied wrench, which can be subject to some uncertainties due to perturbations introduced by external agents. We model such perturbations as a six-dimensional ellipsoid $\mathcal{W}$ centered in $\hat{\boldsymbol{w}}_0$, defined as

$$(\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_0)^\mathsf{T} \boldsymbol{E}\, (\hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_0) \leq 1,$$

where $\boldsymbol{E}$ is a constant 6×6 positive-definite symmetric matrix, and where $\hat{\boldsymbol{w}}$ and $\hat{\boldsymbol{w}}_0$ are assumed to be given in a frame $PXYZ$ centered in P and parallel to $OXYZ$, although any other frame could be assumed if desired. In practice, the ellipsoid $\mathcal{W}$ can be constructed by propagating known bounds on other variables related to $\hat{\boldsymbol{w}}$, using the tools from ellipsoidal calculus [14], for example.

B. Feasibility conditions

The configurations of the *FlyCrane* system that are quasi-statically reachable have to satisfy some feasibility conditions, which are of two types. On the one hand, the system must be able to statically counteract the forces applied on the platform with positive and acceptable cable tensions. Such condition imposes the so-called *wrench-feasibility* constraints on the configurations, which are acquired from cable-driven manipulators [6, 4]. On the other hand, the forces applied on each flying robot cannot exceed the thrust that it is able to exert. Let us formally describe both types of conditions:

The wrench-feasibility requirement on a configuration $\boldsymbol{q} \in \mathcal{C}$ implies that for each wrench $\hat{\boldsymbol{w}} \in \mathcal{W}$ there must be a vector

$$\boldsymbol{f} = [f_1, \ldots, f_6]^\mathsf{T} \in \mathcal{D} = (\underline{f_1}, \overline{f_1}) \times \ldots \times (\underline{f_6}, \overline{f_6})$$

of cable tensions satisfying

$$\boldsymbol{J}(\boldsymbol{q}) \cdot \boldsymbol{f} = \hat{\boldsymbol{w}},$$

where $\boldsymbol{J}(\boldsymbol{q})$ is the 6×6 screw Jacobian of the towed-cable system at $\boldsymbol{q}$ given in the same frame as $\hat{\boldsymbol{w}}$, and $(\underline{f_i}, \overline{f_i})$ is the range of cable tensions that can be resisted by the ith cable, with $\underline{f_i} > 0$. Let $\boldsymbol{f}_0$ be the vector of cable tensions corresponding to a given $\hat{\boldsymbol{w}}_0 \in \mathcal{W}$, i.e.,

$$\boldsymbol{J}(\boldsymbol{q}) \cdot \boldsymbol{f}_0 = \hat{\boldsymbol{w}}_0. \tag{3}$$

By noting that $\boldsymbol{J}(\boldsymbol{q})(\boldsymbol{f} - \boldsymbol{f}_0) = \hat{\boldsymbol{w}} - \hat{\boldsymbol{w}}_0$, it is easy to see that, for a given $\boldsymbol{q}$, the set $\mathcal{F}(\boldsymbol{q})$ of cable tensions $\boldsymbol{f}$ corresponding to all $\hat{\boldsymbol{w}} \in \mathcal{W}$ is the ellipsoid given by

$$(\boldsymbol{f} - \boldsymbol{f}_0)^\mathsf{T} \boldsymbol{K}\, (\boldsymbol{f} - \boldsymbol{f}_0) \leq 1,$$

where $\boldsymbol{K} = \boldsymbol{J}(\boldsymbol{q})^\mathsf{T} \boldsymbol{E}\, \boldsymbol{J}(\boldsymbol{q})$. This ellipsoid will be bounded in all directions or unbounded in some, depending on whether $\det(\boldsymbol{J}(\boldsymbol{q})) \neq 0$ or not. However, it is not difficult to see that $\boldsymbol{J}(\boldsymbol{q})$ is non-singular over the wrench-feasible configurations [4], so that $\mathcal{F}(\boldsymbol{q})$ will always be a bounded ellipsoid in our case. Now, for $\boldsymbol{q}$ to be wrench-feasible, we must have $\mathcal{F}(\boldsymbol{q}) \subseteq \mathcal{D}$, which can be checked as follows. For each $i = 1, \ldots, 6$ let $\boldsymbol{v}_i \in \mathbb{R}^6$ be a vector satisfying

$$\left.\begin{aligned} \boldsymbol{v}_i^\mathsf{T} \boldsymbol{K}\, \boldsymbol{v}_i &= 1 \\ \boldsymbol{K}^i \boldsymbol{v}_i &= \boldsymbol{0} \end{aligned}\right\}, \tag{4}$$

where $\boldsymbol{K}^i$ stands for the matrix $\boldsymbol{K}$ with its ith row removed. Observe that if $\boldsymbol{J}(\boldsymbol{q})$ is non-singular, then $\boldsymbol{K}$ and $\boldsymbol{K}^i$ are full row rank, and if the ith component of $\boldsymbol{v}_i$, $v_{i,i}$, is chosen positive, then there is exactly one vector $\boldsymbol{v}_i$ satisfying Eq. (4). Using Lagrange multipliers, it can be shown that, for the solutions $\boldsymbol{v}_i$ of Eq. (4) with $v_{i,i} \geq 0$, $\boldsymbol{f}_0 - \boldsymbol{v}_i$ and $\boldsymbol{f}_0 + \boldsymbol{v}_i$ are the vectors in $\mathcal{F}(\boldsymbol{q})$ attaining the smallest and largest value along the ith coordinate. Thus, for any configuration $\boldsymbol{q}$, the tensions associated to the ith cable will take values between $\underline{t_i}(\boldsymbol{q}) = \boldsymbol{f}_{0,i} - \boldsymbol{v}_{i,i}$ and $\overline{t_i}(\boldsymbol{q}) = \boldsymbol{f}_{0,i} + \boldsymbol{v}_{i,i}$. Hence, when $\det(\boldsymbol{J}(\boldsymbol{q})) \neq 0$, we have that $\mathcal{F}(\boldsymbol{q}) \subseteq \mathcal{D}$ if, and only if, for $i = 1, \ldots, 6$

$$\underline{t_i}(\boldsymbol{q}) > \underline{f_i}, \tag{5}$$

and

$$\overline{t_i}(\boldsymbol{q}) < \overline{f_i}. \tag{6}$$

Then, the configurations satisfying the conditions of Eqs. (1)-(6) are able to equilibrate any external wrench in $\mathcal{W}$ applied on the platform ensuring that the cables will not be too tight nor slack, i.e. are wrench-feasible.

As we said above, the thrust conditions also need to be satisfied. Clearly, each flying robot is subject to the forces applied by the attached cables i and j and its weight vector $\boldsymbol{g}_{ij}$, whose resultant should not exceed in norm the maximum thrust $h_{ij} > 0$ that the robot is able to exert. Now, if $\boldsymbol{u}_i = \frac{\boldsymbol{b}_i - \boldsymbol{a}_i}{\|\boldsymbol{b}_i - \boldsymbol{a}_i\|}$ is the unit vector associated to the ith cable, then, in order to compensate the applied forces, each robot must satisfy

$$\max\{\|\boldsymbol{g}_{ij} + (f_{0,i} + \lambda_i)\boldsymbol{u}_i + (f_{0,j} + \lambda_j)\boldsymbol{u}_j\|\} < h_{ij}, \tag{7}$$

where $(\lambda_i, \lambda_j) \in [-v_{i,i}, v_{i,i}] \times [-v_{j,j}, v_{j,j}]$.

All the previous conditions define the feasible configurations that the aerial towed-cable system can reach, satisfying both the wrench-feasibility and thrust constraints.

IV. Path planning strategy

The current aim of the *FlyCrane* system is the 6-D quasi-static manipulation of a load. The resolution of such a manipulation problem can be seen as a path-planning query with the additional feasibility constraints given in Section III-B. In fact, the required manipulation motion should also avoid solutions that may approach the violation of such constraints. With this in mind, we will define a quality measure on the configurations $\boldsymbol{q}$ of the system, given as a function $c : \mathcal{C} \to \mathbb{R}_+$, or cost function.

Given two feasible configurations $\boldsymbol{q}_{init}$ and $\boldsymbol{q}_{goal}$, classical sampling-based path planners, such as the Rapidly-exploring Random Tree (RRT) algorithm [11], aim at finding a collision-free, feasible path between them, but are not able to consider a continuous cost function defined over the configuration space. Therefore, we cannot expect to obtain good-quality results with RRT. Instead, we will base our path-planning strategy on a variant of RRT, called the Transition-based RRT (T-RRT) algorithm [9], that takes this cost into account during the configuration-space exploration and that tends to produce a good-quality path, i.e. a path following low-cost regions of the configuration space.

A. Quality measure

Let us first define the criteria that will characterize good-quality configurations, and the function to measure such quality. The quality measure should evaluate whether a feasible configuration is close to, or far from, non-feasible ones. Therefore, a meaningful way to measure this quality relies on the fulfillment of the feasibility constraints provided in Section III, for which we will combine the conditions given in Eqs. (5)-(7).

Given a configuration $\boldsymbol{q}$, we define the cost of $\boldsymbol{q}$, $c(\boldsymbol{q})$, as

$$\frac{1}{\prod_{ij}\left(m_{ij}(\boldsymbol{q}) - h_{ij}\right)\prod_k\left(\underline{t_k}(\boldsymbol{q}) - \underline{f_k}\right)\left(\overline{f_k} - \overline{t_k}(\boldsymbol{q})\right)}, \tag{8}$$

where $m_{ij}(\boldsymbol{q})$ is the maximum value of the left term of Eq. (7) associated to $\boldsymbol{q}$. It is clear that $c(\boldsymbol{q}) > 0$ on any feasible configuration $\boldsymbol{q}$. But whenever some cable tensions approach their limits or whenever the forces applied on some robot approach the thrust of the robot, then $c(\boldsymbol{q})$ tends to infinity. Actually, $c(\boldsymbol{q})$ takes higher values when $\boldsymbol{q}$ gets closer to violate any of the conditions of Eqs. (5)-(7), which is the kind of quality measure that we are looking for. Indeed, we will say that $\boldsymbol{q}_a$ is of better quality than $\boldsymbol{q}_b$ if $c(\boldsymbol{q}_a) < c(\boldsymbol{q}_b)$.

Appendix A shows that $c(\boldsymbol{q})$ is a continuous differentiable function over the set of feasible configurations, which is a crucial property for the T-RRT planner to perform properly, because no abrupt cost changes are expected to occur. It is important to add that while the path is computed in $\mathcal{C} = SE(3) \times \mathbb{T}^3$, it will have to be translated to the space $\mathbb{R}^9$ of quadrotor coordinates to be executed. Since the wrench-feasibility constraints are fulfilled all along the path, the Jacobian $J(\boldsymbol{q})$ of the system will never be singular on it, guaranteeing that the path in $\mathbb{R}^9$ will correspond to a unique smooth path in $\mathcal{C}$. In other words, despite the system being actuated by moving the quadrotors, its stiffness will never be lost, because non-smoothnesses or path bifurcations will never be encountered.

Note finally that the lower-cost regions of $\mathcal{C}$ are very favorable to perform manipulation tasks, not only because they correspond to feasible regions, but also because they maintain a security margin from the configurations where constraint violations occur (either the loss of tension, or the breakage of a cable, or a thrust insufficiency). Finding a path with low cost values will thus be beneficial to properly maneuver the *FlyCrane* system.

B. Transition-based RRT

The principle of RRT is to iteratively construct a tree that tends to rapidly expand on the configuration space, thanks to the implicit enforcement of a Voronoi bias [11]. At each iteration of the tree construction, a configuration $\boldsymbol{q}_{rand}$ is randomly sampled in $\mathcal{C}$, and an expansion toward $\boldsymbol{q}_{rand}$ is attempted, starting from its nearest neighbor in the tree, $\boldsymbol{q}_{near}$, which potentially leads to the addition of a new configuration $\boldsymbol{q}_{new}$ to the tree. T-RRT extends RRT by integrating a stochastic transition test enabling it to steer the exploration toward low-cost regions of the space. This transition test is based on the Metropolis criterion typically used in Monte Carlo optimization methods [16]. These techniques aim at finding global minima in complex spaces and involve randomness as a means to avoid being trapped in local minima. Similarly, T-RRT uses a transition test to accept or reject a candidate

Algorithm 1: Transition-based RRT

```
input : the configuration space $\mathcal{C}$
        the cost function $c : \mathcal{C} \rightarrow \mathbb{R}_+$
        the root configuration $\boldsymbol{q}_{init}$
        the goal configuration $\boldsymbol{q}_{goal}$
output: the tree $\mathcal{T}$
$\mathcal{T} \leftarrow$ initTree($\boldsymbol{q}_{init}$)
while not stopCondition($\mathcal{T}$, $\boldsymbol{q}_{goal}$) do
    $\boldsymbol{q}_{rand} \leftarrow$ sampleRandomConfiguration($\mathcal{C}$)
    $\boldsymbol{q}_{near} \leftarrow$ findNearestNeighbor($\mathcal{T}$, $\boldsymbol{q}_{rand}$)
    if refinementControl($\mathcal{T}$, $\boldsymbol{q}_{near}$, $\boldsymbol{q}_{rand}$) then
        $\boldsymbol{q}_{new} \leftarrow$ extend($\boldsymbol{q}_{near}$, $\boldsymbol{q}_{rand}$)
        if $\boldsymbol{q}_{new} \neq null$
        and transitionTest($\mathcal{T}$, $c(\boldsymbol{q}_{near})$, $c(\boldsymbol{q}_{new})$) then
            addNewNodeAndEdge($\mathcal{T}$, $\boldsymbol{q}_{near}$, $\boldsymbol{q}_{new}$)
```

Algorithm 2: transitionTest ($\mathcal{T}$, c_i, c_j)

```
input : the cost threshold $c_{max}$
        the current temperature $T$
        the temperature increase rate $T_{rate}$
output: true if the transition is accepted, and false otherwise
if $c_j > c_{max}$ then return False
if $c_j \leq c_i$ then return True
if $exp(-(c_j - c_i) \,/\, T) > 0.5$ then
    $T \leftarrow T \,/\, 2^{(c_j - c_i) \,/\, (0.1 \,\cdot\, \text{costRange}(\mathcal{T}))}$
    return True
else
    $T \leftarrow T \cdot 2^{T_{rate}}$
    return False
```

Algorithm 3: refinementControl ($\mathcal{T}$, q_{near}, q_{rand})

```
input : the extension step-size $\delta$
        the refinement ratio $\rho$
output: true if refinement is not too high, and false otherwise
if distance($q_{near}, q_{rand}$) $< \delta$
and nbRefinementNodes($\mathcal{T}$) $> \rho \cdot$ nbNodes($\mathcal{T}$) then
    return False
return True
```

state, based on the cost variation associated with the local motion from the previous state to this state. The pseudocode of T-RRT (shown in Algorithm 1) is similar to that of RRT [11], with the addition of the `transitionTest` and `refinementControl` functions.

The `transitionTest` presented in Algorithm 2 is used to evaluate the transition between the configurations $\boldsymbol{q}_{near}$ and $\boldsymbol{q}_{new}$ based on their respective costs. Three cases are possible: 1) A new configuration whose cost is higher than the threshold value c_{max} is automatically rejected. 2) A transition corresponding to a downhill move ($c_j \leq c_i$) is always accepted. 3) Uphill transitions are accepted or rejected based on the probability $\exp(-(c_j - c_i) \,/\, T)$, which decreases exponentially with the cost variation $c_j - c_i$, similarly to the Metropolis criterion. In that case, the level of difficulty of the transition test is controlled by the adaptive parameter T, called *temperature* here only by analogy to statistical physics. Low temperatures limit the expansion to gentle slopes, and high temperatures enable to climb steep slopes. The temperature is dynamically tuned during the search process, which allows T-RRT to automatically balance its bias toward low-cost regions with the Voronoi bias of RRT. After each accepted uphill transition, T is decreased to avoid over-exploring high-cost regions: More precisely, T is divided by $2^{(c_j - c_i) \,/\, (0.1 \,\cdot\, \text{costRange}(\mathcal{T}))}$, where `costRange(`$\mathcal{T}$`)` is the cost difference between the highest-cost configuration and the lowest-cost configuration present in the tree $\mathcal{T}$. After each rejected uphill transition, T is increased to facilitate exploration and to avoid being trapped in a local minimum: More precisely, T is multiplied by $2^{T_{rate}}$, where $T_{rate} \in \,]0, 1]$ is the temperature increase rate. In the rest of the paper, we use $T_{rate} = 0.1$ and we initialize T to 10^{-6}. A value can be provided for c_{max} only when prior knowledge of the planning problem is available and some regions of the space are forbidden. Note that, in the space where configurations whose cost is greater than c_{max} are considered as part of the obstacle regions, T-RRT is probabilistically complete [9].

The adaptive temperature tuning of T-RRT ensures a given success rate for uphill transitions, which can also contribute to refining the exploration of low-cost regions already reached by the tree, as a side effect. The objective of the `refinementControl` function (shown in Algorithm 3) is to limit this refinement and facilitate the tree expansion toward unexplored regions. The idea is to reject an expansion that would lead to more refinement if the ratio of current refinement nodes with respect to the total number of nodes in the tree is greater than a certain value ρ, a refinement node being defined as a node whose distance to its parent is less than the extension step-size δ. Another benefit of the refinement control is to limit the number of nodes in the tree and thus to reduce the computational cost of the neighbor search. Following the suggestion in [9], we set ρ to 0.1.

V. TEST CASES

In this section, we evaluate the proposed approach on two 6-D quasi-static manipulation problems involving the *FlyCrane* system (cf. Fig. 1). The first example is a complex task (inspired by classical motion planning benchmarks) in which the *FlyCrane* has to get a 3-D puzzle piece through a hole, as illustrated by Fig. 4. The second example, presented in Fig. 5, simulates a more realistic situation in which the *FlyCrane* has to install a lightweight footbridge between two buildings to evacuate people during a rescue operation. These examples differ in terms of difficulty: the *Rescue* problem is the easiest one because it requires only a translation and two rotations about a single axis of the *FlyCrane* for a solution to be found; the *Puzzle* problem requires to simultaneously perform a translation and four rotations about two axes of the *FlyCrane*. In both problems, the *octahedral FlyCrane* with an equilateral platform is considered. In our simulation environment, the cables are 2 m long, and the platform sides are 1.4 m long.

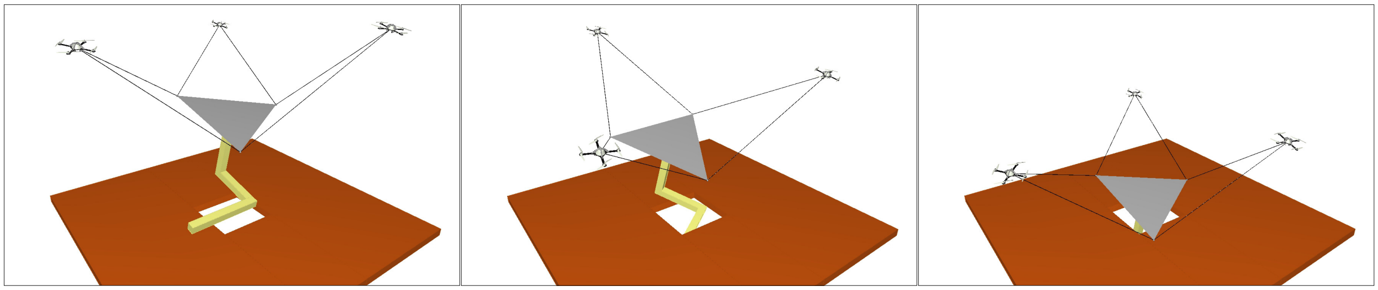

Fig. 4. The *Puzzle* problem: the *FlyCrane* has to get a 3D puzzle piece through a hole.

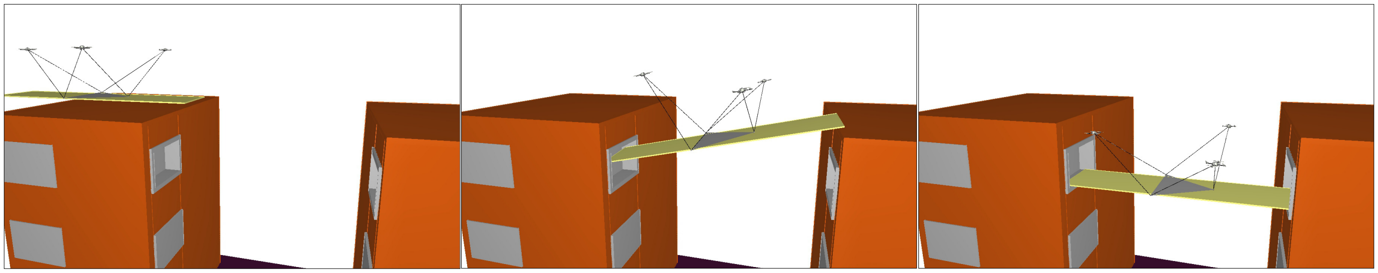

Fig. 5. The *Rescue* problem: the *FlyCrane* has to install a lightweight footbridge between two buildings for a rescue operation.

TABLE I

EVALUATION OF RRT AND T-RRT ON THE *Puzzle* AND *Rescue* PROBLEMS. AVERAGE VALUES OVER 100 RUNS ARE GIVEN FOR: THE AVERAGE COST $avgC$, THE MAXIMAL COST $maxC$, THE MECHANICAL WORK MW, THE INTEGRAL OF THE COST IC, THE RUNNING TIME t (IN SECONDS), THE NUMBER OF NODES N IN THE TREE, AND THE NUMBER OF EXPANSION ATTEMPTS X.

	Puzzle							*Rescue*						
	$avgC$	$maxC$	MW	IC	t (s)	N	X	$avgC$	$maxC$	MW	IC	t (s)	N	X
RRT	1130	11,684	11,651	300,793	34	2654	15,609	102	575	554	80,750	126	1361	193,517
T-RRT	78	229	193	30,352	169	4698	78,501	36	42	11	24,588	54	379	207,778

On both examples, we evaluate the performance of the RRT and T-RRT algorithms on the basis of their running time t (in seconds), the number of attempted expansions X, and the number of nodes N in the produced tree. To avoid generating trivially-non-feasible paths, RRT only accepts feasible (i.e. collision-free and satisfying the aforementioned feasibility constraints) configurations. After performing a smoothing operation (based on the random shortcut method [8]) on the paths generated by RRT and T-RRT, we evaluate the path quality by computing the average cost $avgC$, the maximal cost $maxC$, the mechanical work MW, and the integral of the cost IC. The *mechanical work* of a path is the sum of the positive cost variations along the path [9]. For all variables, we give values averaged over 100 runs, as reported in Table I.

Unsurprisingly, Table I shows that T-RRT provides better-quality paths than RRT on both examples: on the *Puzzle* problem, all cost statistics are more than one order of magnitude lower for paths generated by T-RRT; on the *Rescue* problem, they are between three and 50 times lower. Since it generally requires more expansion attempts to find configurations with acceptable cost, T-RRT is often slower than RRT, as is the case on the *Puzzle* problem (169 s vs 34 s). However, it is worth noting that T-RRT runs faster than RRT on the *Rescue* problem (54 s vs 126 s), thanks to the lower number of nodes added to the tree (379 vs 1361), which makes the nearest-neighbor search faster.

We were interested in finding out what made path quality differ between RRT and T-RRT. For that, we computed the tensions exerted on each cable and the forces exerted on each quadrotor, along the paths produced by RRT and T-RRT, after dividing every path into 100 steps corresponding to intermediate configurations of the system. Then, for each path-step, we computed the minimal and maximal tensions (over all cables) and forces (over all quadrotors) over the 100 paths produced by RRT and over the 100 paths produced by T-RRT. Therefore, for each step, we obtained the tension ranges and the force ranges yielded by RRT and T-RRT. Fig. 6 presents the profiles of the tension range and of the force range, respectively, for the *Rescue* problem. Similar plots have been obtained for the *Puzzle* problem. We can see that using T-RRT leads to narrower tension and force ranges than using RRT. Most importantly, we observe that RRT produces paths along which a tension or a force can be dangerously close to a bound of its validity interval. For example, Fig. 6.a shows that, along some path, at least one tension comes close to zero, meaning that at least one cable almost goes slack.

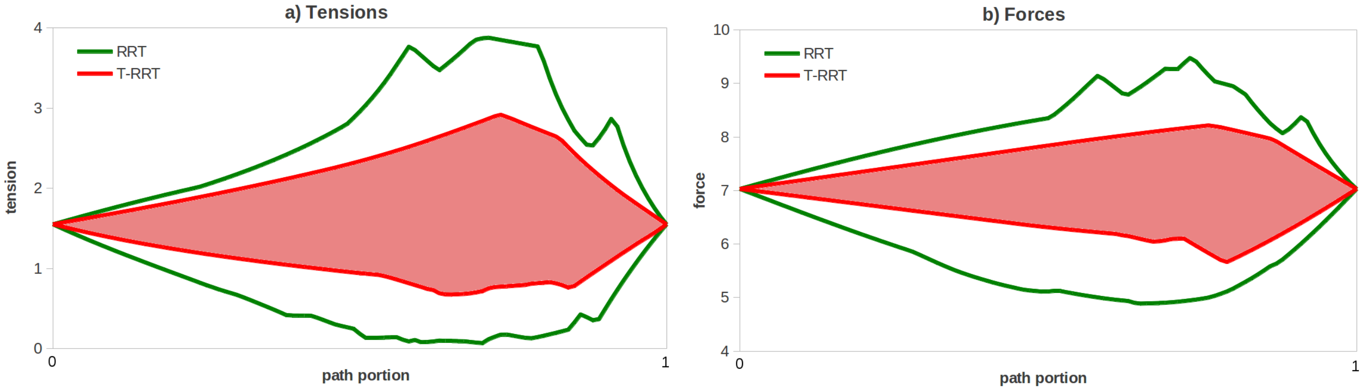

Fig. 6. Profiles of a) the tension range and b) the force range, observed over 100 paths produced by RRT and T-RRT on the *Rescue* problem. The filled areas between the red curves represent the ranges for T-RRT; the areas between the green curves represent the ranges for RRT.

Similarly, on the *Puzzle* problem, one force comes close to the maximal thrust value. As a conclusion, we argue that integrating the path-planning T-RRT algorithm into the 6-D proposed manipulation approach allows us to plan safer paths for the *FlyCrane* system.

VI. Conclusion

We have presented an approach for the 6-dimensional quasi-static manipulation of a load with an aerial towed-cable system. The main contribution of the approach lies in the combination of results deriving from the static analysis of cable-driven manipulators with the application of a cost-based motion-planning algorithm to solve manipulation queries. The link underlying this combination is the definition of a quality measure for the configurations of the system. First, this quality measure is based on the wrench-feasibility constraints applied to cable-driven manipulators and on additional thrust constraints, and allows: 1) to discriminate non-feasible from feasible configurations, and 2) to favor configurations that are far from violating these constraints, by attributing them a low cost. Second, this quality measure leads to the definition of a cost function, thus allowing for the use of a cost-based motion-planning algorithm, namely the Transition-based RRT (T-RRT). As a result, rather than simply computing collision-free paths, the proposed approach produces good-quality paths, with respect to the constraints imposed on the system.

As part of our approach, we have additionally proposed an aerial towed-cable system that we have named the *FlyCrane*. This system consists of a platform attached to three flying robots by means of three pairs of fixed-length cables. We have evaluated the approach, in simulation, on two 6-D manipulation problems involving an octahedral version of the *FlyCrane* system. The results of the evaluation show that the proposed motion planning approach is suitable to solve 6-D quasi-static manipulation tasks. Furthermore, they have confirmed that RRT, which is the original variant of T-RRT that does not take the cost into account, may produce paths that occasionally approach dangerous situations, while T-RRT produces safer paths.

The proposed approach allows for extensions in several ways. In particular, we expect to extend the method to consider positioning errors for the flying robots, which could be due to external force perturbations and to errors in the localization methods. For this, similar techniques to those applied in this paper could be used. Additionally, an interesting and challenging extension to this work is the introduction of dynamics in the motion of the load and of the flying robots, as they play an important role in the overall manipulation of the system.

In this paper, we have applied the proposed approach in simulated environments. As part of our future work, we plan to implement this approach in a real aerial towed-cable system. This will serve as a testbed for the validation of the method and its further extensions, providing relevant feedback on the real limitations of the approach and the system. In real-life situations, the proposed approach could be helpful in various applications. As illustrated by the simulated *Rescue* problem, one possible application is the construction of platforms for the evacuation of people in rescue operations. Another application could be the installation of platforms in uneven terrains for the landing of manned or unmanned aircrafts. More generally, it could be useful for the assembly of structures in places difficult to access for humans.

Acknowledgments

This work has been partially supported and by the Spanish Ministry of Economy and Competitiveness under contract DPI2010-18449, by the European Community under contract ICT 287617 “ARCAS”, and by a *Juan de la Cierva* contract supporting the first author.

References

[1] J. Albus, R. Bostelman, and N. Dagalakis. The NIST Robocrane. *J. Robotic Systems*, 10(5):709–724, 1993.

[2] D. Berenson, T. Siméon, and S. Srinivasa. Addressing Cost-Space Chasms in Manipulation Planning. In *IEEE Int. Conf. Robotics and Automation*, pages 4561–4568, 2011.

[3] M. Bernard, K. Kondak, I. Maza, and A. Ollero. Autonomous transportation and deployment with aerial robots for search and rescue missions. *J. Field Robotics*, 28(6):914–931, 2011.

[4] O. Bohigas, M. Manubens, and L. Ros. Navigating the Wrench-Feasible C-Space of Cable-Driven Hexapods. In *Cable-Driven Parallel Robots*, pages 53–68. Springer, 2012.

[5] I.A. Bonev, D. Zlatanov, and C.M. Gosselin. Advantages of the modified Euler angles in the design and control of PKMs. In *Parallel Kinematic Machines Int. Conf.*, pages 171–188, 2002.

[6] P. Bosscher, A.T. Riechel, and I. Ebert-Uphoff. Wrench-feasible workspace generation for cable-driven robots. *IEEE Trans. Robotics*, 22(5):890–902, 2006.

[7] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *Int. J. Robotics Research*, 30(3):324–334, 2011.

[8] R. Geraerts and M. H. Overmars. Creating high-quality paths for motion planning. *Int. J. Robotics Research*, 26 (8):845–863, 2007.

[9] L. Jaillet, J. Cortés, and T. Siméon. Sampling-based path planning on configuration-space costmaps. *IEEE Trans. Robotics*, 26(4):635–646, 2010.

[10] L. Jaillet, F.J. Corcho, J.J. Pérez, and J. Cortés. Randomized tree construction algorithm to explore energy landscapes. *J. Computational Chemistry*, 32(16):3464–3474, 2011.

[11] S. M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308. A K Peters, 2001.

[12] N. Michael, J. Fink, and V. Kumar. Cooperative manipulation and transportation with aerial robots. *Autonomous Robots*, 30(1):73–86, 2011.

[13] R. M. Murray. Trajectory Generation for a Towed Cable System Using Differential Flatness. In *IFAC World Congress*, 1996.

[14] L. Ros, A. Sabater, and F. Thomas. An ellipsoidal calculus based on propagation and fusion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 32(4):430–442, 2002.

[15] R. A. Skop and Y. I. Choo. The Configuration of a Cable Towed in a Circular Path. *J. Aircraft*, 8:856–862, 1971.

[16] J.C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, 2003.

[17] A. Steltzner, D. Kipp, A. Chen, D. Burkhart, C. Guernsey, G. Mendeck, R. Mitcheltree, R. Powell, T. Rivellini, M. San Martin, and D. Way. Mars Science Laboratory entry, descent, and landing system. In *IEEE Aerospace Conf.*, 2006.

[18] P. Williams. Optimal terrain-following for towed-aerial-cable sensors. *Multibody System Dynamics*, 16(4):351–374, 2006.

Appendix

Let us prove that the cost $c(\boldsymbol{q})$, as defined in Sec. IV-A, is a continuously differentiable function $c : \mathcal{C} \to \mathbb{R}_+$. Note that $c(\boldsymbol{q})$ can be equivalently determined as the solution $c > 0$ of

$$c \prod_{ij} (m_{ij}(\boldsymbol{q}) - h_{ij}) \prod_k (\underline{t_k}(\boldsymbol{q}) - \underline{f_k})(\overline{f_k} - \overline{t_k}(\boldsymbol{q})) = 1. \quad (9)$$

We can now define a system of polynomial equations formed by Eqs. (1)-(4), and Eq. (9), which we write as

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}, \quad (10)$$

where $\boldsymbol{x}$ refers to an n_x-vector encompassing all of its variables, $(\boldsymbol{q}, \boldsymbol{R}, \boldsymbol{a}_i, \boldsymbol{f}_0, \boldsymbol{v}_i, c)$. By the *Implicit Function Theorem*, the solutions c of Eq. (10) will define a continuously differentiable function $c(\boldsymbol{q})$ if the differential matrix of $\boldsymbol{F}$ with respect to all variables $\boldsymbol{x}$ except those of $\boldsymbol{q}$, $\boldsymbol{F}_{\{\boldsymbol{x}-\boldsymbol{q}\}}$, is full rank. Such matrix is of the form

$$\left(\begin{array}{c|c|c|ccc|c}
-\boldsymbol{I}_3 & & & & & & \\ \hline
* & -\boldsymbol{I}_3 & & & & & \\ \hline
\multicolumn{2}{c|}{*} & \boldsymbol{J}(\boldsymbol{q}) & & & & \\ \hline
 & & & \begin{matrix}2\boldsymbol{v}_1^\top\boldsymbol{K}\\ \boldsymbol{K}^1\end{matrix} & & & \\
 & & * & & \ddots & & \\
 & & & & & \begin{matrix}2\boldsymbol{v}_6^\top\boldsymbol{K}\\ \boldsymbol{K}^6\end{matrix} & \\ \hline
 & & \multicolumn{4}{c|}{*} & d
\end{array}\right),$$

where $d = \prod_{ij}(m_{ij}(\boldsymbol{q}) - h_{ij}) \prod_k (\underline{t_k}(\boldsymbol{q}) - \underline{f_k})(\overline{f_k} - \overline{t_k}(\boldsymbol{q}))$, the empty blocks represent zero-matrices, and asterisks indicate non-zero blocks. For being a lower triangular square matrix, it is sufficient to prove that the diagonal blocks are all full rank. Clearly, the 3×3 identity matrices are full rank.

The screw Jacobian $\boldsymbol{J}(\boldsymbol{q})$ can be shown to be full rank over the wrench-feasible configurations by contradiction. Indeed, if $\boldsymbol{J}(\boldsymbol{q}_s)$ were rank deficient for some $\boldsymbol{q}_s$, then so would be $\boldsymbol{K}$. In such case, there would exist some i for which all $\boldsymbol{v}_i \in \ker(\boldsymbol{K}^i)$ would lie in $\ker(\boldsymbol{K})$, and thus, it would be $\boldsymbol{v}_i^\top \boldsymbol{K} \boldsymbol{v}_i = 0$, which contradicts Eq. (4), and hence $\boldsymbol{J}(\boldsymbol{q})$ cannot be rank deficient over the feasible configurations.

The 6×6 block matrices involving $\boldsymbol{K}$ and $\boldsymbol{K}^i$ can only be rank deficient if $v_{i,i} = 0$, which cannot be satisfied along the feasible configurations, as we are about to prove. If $v_{i,i} = 0$ for some i, then by replacing $\boldsymbol{K}^i \boldsymbol{v}_i = \boldsymbol{0}$ into $\boldsymbol{v}_i^\top \boldsymbol{K} \boldsymbol{v}_i = 1$, we obtain the dot product of two vectors: $\boldsymbol{v}_i^\top$, with $v_{i,i} = 0$, and the vector $\boldsymbol{K}\boldsymbol{v}_i$, whose components are all zero except that in position i. The result of this dot product is 0, which contradicts Eq. (4), as it should be 1.

Finally, d is never zero, because of Eq. (9). Therefore, we have shown the existence and the continuous differentiability of function c. Additionally, as the previous matrix $\boldsymbol{F}_{\{\boldsymbol{x}-\boldsymbol{q}\}}$ is a full-rank sub-matrix of maximum dimension of the Jacobian $\boldsymbol{F}_{\{\boldsymbol{x}\}}$, we can also certify that the solution set of Eq. (10) is a smooth manifold.