Optimization and Engineering manuscript No.
(will be inserted by the editor)

Computing the Lowest Equilibrium Pose of a Cable-Suspended Rigid Body

Jean-François Collard · **Philippe Cardou**

Received: date / Accepted: date

Abstract We solve the problem of finding the lowest stable-equilibrium pose of a rigid body subjected to gravity and suspended in space by an arbitrary number of cables. Besides representing a contribution to fundamental rigid-body mechanics, this solution finds application in two areas of robotics research: underconstrained cable-driven parallel robots and cooperative towing. The proposed approach consists in globally minimizing the rigid-body potential energy. This is done by applying a branch-and-bound algorithm over the group of rotations, which is partitioned into boxes in the space of Euler-Rodrigues parameters. The lower bound on the objective is obtained through a semidefinite relaxation of the optimization problem, whereas the upper bound is obtained by solving the same problem for a fixed orientation. The resulting algorithm is applied to several examples drawn from the literature. The reported Matlab implementation converges to the lowest stable equilibrium pose generally in a few seconds for cable-robot applications. Interestingly, the proposed method is only mildly sensitive to the number of suspending cables, which is shown by solving an example with 1000 cables in two hours.

Keywords Cable-suspended rigid body · cable robot · parallel robot · branch-and-bound algorithm · semidefinite programming.

J.F. Collard
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier,
Université Montpellier II – C.N.R.S. 161, rue Ada, 34095 Montpellier Cedex 5, France,
Tel.: +33-467-418559, Fax: +33-467-418500
E-mail: jfr.collard@gmail.com

P. Cardou
Laboratoire de robotique,
Université Laval, Quebec City, QC, Canada, G1V 0A6,
E-mail: pcardou@gmc.ulaval.ca

1 Introduction

The problem addressed in this paper is that of finding the lowest equilibrium pose (orientation and reference-point position) of a rigid body subjected to gravity, and attached to a fixed frame by n cables whoses lengths are known. By *height of a pose*, we refer to the height of the centre of mass of the rigid body under consideration, for the given pose. Although finding this pose may be regarded as a fundamental problem of rigid-body mechanics, to our knowledge, it was never solved. Not without reason: when considering generic, spatial, rigid-body and fixed-frame geometries, the number of equilibrium poses quickly explodes as the number of cables increases. This combinatorial behaviour makes an algebraic approach to the problem very difficult in general.

On the other hand, recent reports indicate that producing an algorithm that would adequately solve for the equilibrium pose of a cable-suspended rigid body would have an impact on at least two effervescent fields of robotics: cable-driven parallel manipulators (Ghasemi et al 2010; Carricato and Merlet 2010) and cooperative towing (Michael et al 2009; Fink et al 2009).

1.1 Cable-Driven Parallel Manipulators

In the first field, the problem may be termed *the forward displacement analysis of underconstrained cable-driven parallel manipulators*. It is closely related to the famous problem of the forward kinematics of the Gough-Stewart platform, in which one seeks the moving-platform pose, given the lengths of the six actuated legs. The solution to this problem is relatively recent (Wampler 1996; Husty 1996), which gives a clue of its complexity. One important distinction between the two problems is that the equality constraint associated with each leg of the Gough-Stewart platform becomes an inequality constraint corresponding to a cable of the cable-driven parallel manipulator. In the former case, the intersection of the constraints generally yields a zero-dimensional set of at most 40 solutions, whereas in the latter, there generally is a six-dimensional set of feasible poses containing several locally-optimum solutions, among which the global optimum is the lowest equilibrium pose.

Furthermore, we should point out that the forward displacement analyses of fully-constrained and overconstrained cable-driven parallel manipulators fall outside the scope of this work. By *fully-constrained*, we refer to seven-cable parallel manipulators (e.g., the Falcon Kawamura et al (1995)), whose cables are to remain constantly in tension in order to counter balance any wrench that would be applied to the moving platform. By *overconstrained*, we refer to cable-driven parallel manipulators composed of eight or more cables (e.g., the locomotion interface Perreault and Gosselin (2008)) that remain constantly in tension for the same purpose. Notice that the methods developed for Gough-Stewart platforms can readily be applied to the forward displacement analysis of these types of cable-driven parallel manipulators. Indeed, as

their cables are to remain taut at all times, their geometry is completely analogous to that of a Gough-Stewart platform, thereby allowing the use of the solutions proposed in Wampler (1996) and Husty (1996).

Turning our attention back towards underconstrained cable-driven parallel manipulators, Carricato and Merlet (2011) recently proposed a method for the computation of all stable equilibrium poses of a three-cable manipulator. In this article, they found an upper bound of 156 to the number of stable equilibrium poses admitted by such a mechanism. Their approach consists in first solving the Newton-Euler equilibrium equations of the end effector, and then filtering out the solutions that do not correspond to stable equilibrium poses. As these authors point out, in a stable equilibrium pose, not all cables of the manipulator need to be taut. In fact, any nonempty subset of cables can be under tension, which requires that the equilibrium equations be solved for all possible combinations of cables. As a result, the actual number of equilibrium poses may be high even for this limited number of cables, and, due to the combinatorial nature of the problem, it is expected to explode as the number of cables increases. Hence, we conjecture that the mathematical tools that proved successful in the solution of this latter problem are not suited for the general problem addressed here.

Another recent report (Ghasemi et al 2010) on the forward displacement analysis of underconstrained cable-driven parallel manipulators suggests to use neural networks to solve the associated system of polynomial equations. According to this scheme, the neural network is *trained* by solving the corresponding inverse displacement analysis, which is much easier than the original problem, over a large set of poses. The resulting neural network may provide a good approximation of the forward displacement analysis in many cases. However, it offers no guarantee of convergence to the lowest equilibrium pose, or even to any equilibrium pose in general.

1.2 Cooperative Towing

In the second field of cooperative towing, a payload is suspended in the air by several aerial vehicles. The aerial vehicle displacements are measured, so that the positions of the cable-attachment points are known, as well as the lengths of the cables. The problem consists in finding the pose of the suspended payload. Michael et al (2009) propose a solution to the case where there are $n = 2$ cables (i.e., the planar case). This solution is obtained by finding the equilibrium points on the coupler curve of the analogous planar four-bar linkage. In the same reference, Michael et al. adopt an energetic approach to the case where $n = 3$, using the fact that an equilibrium pose corresponds to a minimum in the potential energy. This leads to a nonconvex optimization problem, whose optima are obtained by varying the initial guess of a local optimization procedure. Such a method, however, does not guarantee that the pose found corresponds to the one where the centre of mass is lowest.

We conjecture that this is the motivation that led the same group of researchers to publish another paper on the topic (Fink et al 2009), where they relax the previously formulated optimization problem into a convex optimization problem. The optimum objective value of the relaxed problem may be regarded as a lower bound on the optimum value of the original problem. Furthermore, the authors provide geometric conditions under which the lower bound is guaranteed to be tight, i.e., under which the optima of the relaxed and original problem coincide.

In a parallel effort, Jiang and Kumar (2010) were able to compute all stable equilibrium poses—including that of lowest centre-of-mass position—for a class of special cases in three-dimensional space. A geometry is a member of this class only if (i) the cable attachment points on the rigid body are located at the vertices of a regular polygon; (ii) the rigid-body centre of mass is at the centroid of the said polygon; (iii) the fixed cable attachment points (i.e., those on the supporting aerial vehicles) form a regular polygon with the same planes of symmetry as the rigid-body polygon; and (iv) this fixed polygon is perpendicular to gravity (i.e., lies in a horizontal plane). These constraints allow for a decomposition of the spatial problem into several planar ones, which are solved by computing the stationary points on the coupler curve of the equivalent four-bar linkage.

A summary of the methods reviewed above is given in Table 1. From this comparison, one notices that no method applies to the completely general case, namely, that no method can compute all equilibrium poses of rigid body suspended by an arbitrary number of cables in an arbitrary geometry. In this paper, we focus on computing the lowest equilibrium pose, while making no particular assumption on the number of cables and geometry. This lowest equilibrium pose provides a tight lower bound on the height of the rigid-body centre of gravity, a piece of information that is useful for guaranteeing no collision while moving a cable-suspended object above obstacles. Moreover, in some applications, it is possible to force the system into this lowest equilibrium pose by controlling the initial conditions. For instance, consider the case where the towing vehicles are required to fly in a specific formation, with given cable lengths and attachment points on the payload. The lowest equilibrium pose of the payload *relative* to the vehicle formation can be computed with the algorithm proposed herein. Thence, it is possible to move the vehicles to their computed relative positions prior to lifting the payload, while it remains still. In this situation, the system is initially in the computed stable-equilibrium pose. Moreover, the system is to remain in this configuration throughout the towing operation, because this pose corresponds to a global minimum in the potential energy, thus guaranteeing that no small perturbation can take the system to another state of equilibrium. Hence, although the existing methods reported in Table 1 have other virtues, none of them solves the general problem posed in this paper, i.e., they do not give a method for computing the lowest equilibrium pose of *any* cable-suspended rigid body.

Table 1 Existing methods for the forward displacement analysis of cable-driven parallel manipulators

method	**num. of cables** m	**geometry**	**equilib. poses**	**comp. time**
Michael et al (2009)	2	any	all	ms
Carricato and Merlet (2011)	3	any	all	NA
Fink et al (2009)	3	constraints	lowest	ms
Jiang and Kumar (2010)	any	symmetric	all	ms
Ghasemi et al (2010)	any	any	any	NA
proposed method	any	any	lowest	s

1.3 The Proposed Approach

In this paper, we proceed very much like Fink et al (2009), in that we resort to a relaxation of the energy-minimization problem. However, the relaxation we obtain is different, and is used for a different purpose. As will be seen, the lower bound provided by the relaxation is cast in a branch-and-bound algorithm, which allows the computation of a global optimum to the energy-minimization problem. This global optimum corresponds to the lowest equilibrium pose of the rigid body. Unlike algebraic approaches, the proposed method does not yield the complete set of equilibrium poses, unless, of course, there is only one such pose. On the other hand, it can guarantee that no solution exists if need be. We also note that the method proposed here does not account for possible interference problems between two cables or between the rigid body and a cable, which is also the case in other existing methods. Nevertheless, we will see that the method proposed here can be applied to problems with large numbers of cables, which sets it apart from the ones that were proposed previously.

This method will be explained in the following order. In section 2, the general energy-minimization problem is formulated. The proposed branch-and-bound algorithm is detailed in sections 3.1, 3.2 and 3.3, where the required partitioning strategy, lower bound and upper bound are defined, respectively. A summary of the algorithm is also provided in section 3.4. Finally, benchmark examples are provided in sections 4.1, 4.2 and 4.3.

2 Problem Formulation

Let us begin by defining the problem as clearly as possible. We are to compute the lowest equilibrium pose of a rigid body suspended by n cables, as depicted in Fig. 1. It is assumed that the mass of the cables is negligeable with respect to the mass m of the suspended body. The sagging effect of the cables is thus not considered here. Moreover, the cables are assumed to be inextensible. In Fig. 1, O is the fixed reference point. The position of the attachment point A_j of the j^{th} cable on the fixed frame is given with respect to O by vector $\mathbf{a}_j$. The other end of the j^{th} cable attaches to the rigid body at B_j. The position of B_j with respect to the rigid-body centre of mass G is given by vector $\mathbf{Qb}_j$, where $\mathbf{b}_j$ represents the position of B_j in a reference orientation of the rigid body, and $\mathbf{Q}$ is a rotation matrix representing its current orientation. In

turn, the position of the centre of mass is given with respect to O by vector $\mathbf{p}$. Hence, the pose of the rigid body is fully represented by $\mathbf{p}$ and $\mathbf{Q}$, which are the variables sought in this paper.

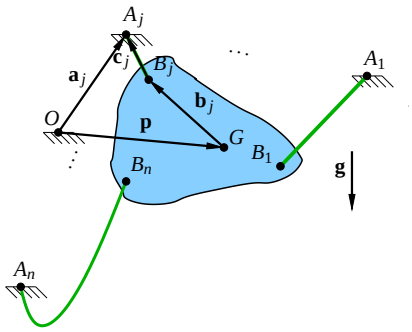

Fig. 1 A rigid body suspended by n cables

The equilibrium pose of the rigid body is constrained by the n cables that attach it. In the proposed model, a cable that is not taut is assumed to exert no force on the rigid body, whereas a taut cable is taken to be completely inextensible, i.e., perfectly rigid under tension forces. In order to represent the cable constraints, let us define vector $\mathbf{c}_j$ pointing from B_j to A_j, which *supports* the j^{th} cable, i.e.,

$$\mathbf{c}_j \equiv \mathbf{a}_j - \mathbf{Q}\mathbf{b}_j - \mathbf{p}. \tag{1}$$

The length of the j^{th} cable is c_j, and is assumed to be known. As a result, we obtain the conditions

$$\|\mathbf{c}_j\|_2 \leq c_j, \ j = 1, \ldots, n, \tag{2}$$

which must be satisfied at all times.

The lowest equilibrium pose of the rigid body may now be expressed as the pose that minimizes its potential energy V, i.e.,

$$\begin{array}{lll} \text{minimize} & V = -m\mathbf{g}^T\mathbf{p}, & \text{(P)} \\ \text{subject to} & \|\mathbf{c}_j\|_2 \leq c_j, \ j = 1, \ldots, n, & \\ \text{over} & \mathbf{p} \in \mathbb{R}^3, \ \mathbf{Q} \in \mathrm{SO}(3), & \end{array}$$

where $\mathrm{SO}(3)$ is the special group of proper orthogonal matrices. This group represents a non-convex set in $\mathbb{R}^{3\times 3}$, and, as a result, the optimization problem (P) is non-convex as well.

Therefore, this problem, which may appear relatively simple at first glance, turns out to be a challenging one. By definition, each stable equilibrium pose corresponds

to a local minimum in the potential energy, and, therefore, finding *any* such pose can be done by applying any descent method to (P). However, descent methods offer no guarantee regarding global optimality, and in this article, we simply regard a local equilibrium as an upper bound on the lowest equilibrium pose. An important advantage of the proposed energetic formulation over the classical Newton-Euler formulation is that it lends itself to the application of the wealth of optimization methods available in the literature.

3 The Proposed Branch-and-Bound Algorithm

To the knowledge of the authors, branch-and-bound algorithms (Land and Doig 1960) are the only procedures that offer guarantees on global optimality while being applicable to any non-convex optimization problem, at least in principle. In the case of problem (P), an alternative would be to solve the associated Karush-Kuhn-Tucker conditions, which could be expressed as polynomial equations in the decision variables and Lagrange multipliers. We rule out this approach, however, as it does not allow to circumvent the combinatorial nature of the problem, and leads to a solution that is similar to that of Carricato and Merlet (2011). We conjecture that this solution would be applicable to systems with three cables at most, and at the cost of much complexity and potentially high computation times—to the knowledge of the authors, no computation times have been reported so far for the solution of a generic three-cable system.

We rather resort to a branch-and-bound method, whose complexity grows mildly with the number of cables. This type of algorithm divides the optimization domain into non-overlapping subsets, over which one is to compute lower and upper bounds on the objective of (P). These bounds should be devised so that shrinking a subset to a point implies that the corresponding bounds converge to the objective-function value at this same point, one from below, the other from above (see Balakrishnan et al (1991)). Taking advantage of this property, the optimization domain is recursively divided until the minimum lower bound and its corresponding upper bound are within a prescribed tolerance of each other. Notice that interval analysis may be regarded as a special case of the branch-and-bound method where the bounds are computed using interval arithmetics. Although this approach may be applicable to problem (P), we rather resort to convex relaxations and semidefinite programming for the computation of the bounds. This choice stems from the conjecture that a specifically-tailored relaxation can outperform interval arithmetics both in accuracy and convergence rate. Moreover, the proposed relaxation methods have been applied with success in other engineering applications such as signal processing and medical imaging (Luo et al 2010).

3.1 Partitioning the Feasible Domain

In general, it is preferable to divide the domain along the fewest possible number of dimensions. In this case, however, we are to represent $\mathrm{SO}(3)$ with the four Euler-Rodrigues (ER) parameters $\mathbf{q} \in \mathbb{R}^4$, that is with one more parameter than the required minimum of three. This choice is based on the observation that the rotation matrix is a quadratic form in the ER parameters, which can be expressed as (see Angeles 2007)

$$\mathbf{Q} = (r_0^2 - \mathbf{r}^T\mathbf{r})\mathbf{1}_{3\times 3} + 2\mathbf{r}\mathbf{r}^T + 2r_0\mathrm{cpm}(\mathbf{r}), \tag{3}$$

where $\mathbf{q} \equiv [r_0 \quad \mathbf{r}^T]^T$, $r_0 \in \mathbb{R}$ and $\mathbf{r} \in \mathbb{R}^3$, $\mathbf{1}_{3\times 3}$ is the three by three identity matrix, and $\mathrm{cpm}(\mathbf{r})$ is the cross-product matrix[1] of $\mathbf{r}$. To this, we adjoin the unit-hypersphere constraint, which is quadratic in $\mathbf{q}$, namely,

$$\mathbf{q}^T\mathbf{q} = 1. \tag{4}$$

Besides the quadratic nature of the expressions in $\mathbf{q}$, another often-cited advantage of ER parameters is that, unlike three-parameter representations of $\mathrm{SO}(3)$, they are free from singularities. However, they entail a double-covering of $\mathrm{SO}(3)$, i.e., a rotation represented by $\mathbf{q}$ is also represented by $-\mathbf{q}$. This property is an obstacle to the application of the branch-and-bound algorithm, as it implies that global optima come in redundant pairs, thereby dividing the convergence rate of the algorithm by two. To resolve this problem, we drop the bottom half of the unit hypersphere, that is, we add the constraint $r_0 \geq 0$. By doing so, we introduce a singularity at $r_0 = 0$, and lose the singularity-free property of these parameters. Nevertheless, for the purpose of this work, the main advantage of ER parameters over other representations of rigid-body rotations remains, namely, they appear quadratically in the constraints (3) and (4).

Hence, we are to partition the four-dimensional space of Euler-Rodrigues parameters into boxes (a.k.a. orthotopes), which we represent as

$$[\underline{\mathbf{q}}, \overline{\mathbf{q}}] \equiv \{\mathbf{q} \in \mathbb{R}^4 : \underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}}\}, \tag{5}$$

where $\leq$ denotes the componentwise inequality. As we choose the top half $r_0 \geq 0$ of the unit hypersphere, then the smallest box containing all possible values of $\mathbf{q}$ is

$$[\underline{\mathbf{q}}, \overline{\mathbf{q}}] = \left[\begin{bmatrix} 0 \\ -\mathbf{1}_3 \end{bmatrix}, \mathbf{1}_4\right], \tag{6}$$

where $\mathbf{1}_n \equiv [1 \quad \cdots \quad 1]^T \in \mathbb{R}^n$. The box of eq. (6) is the initial box of the proposed branch-and-bound algorithm.

[1] The cross-product matrix of $\mathbf{r} \in \mathbb{R}^3$ is defined as $\mathrm{cpm}(\mathbf{r}) \equiv \partial(\mathbf{r} \times \mathbf{x})/\partial\mathbf{x}$ for any $\mathbf{x} \in \mathbb{R}^3$.

3.2 Computing a Lower Bound

Computing the lower and upper bounds on the objective over a given box $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$ is generally the critical step of a branch-and-bound algorithm. In general, good lower and upper bounds are those that are accurate and quickly evaluated. These two criteria are often opposed to one another, so that the best bounds are often obtained by a trade-off. As mentioned previously, accurate upper bounds may be provided by any applicable descent method, whereas accurate lower bounds can be obtained by solving convex relaxations of the original problem. Interval arithmetics is yet another method for obtaining these bounds.

We start with the computation of a lower bound, which is done by performing a semidefinite relaxation of problem (P). The application of semidefinite relaxations (and semidefinite programming in general) to robotics problems seems to have been mentioned only incidentally in previous literature. Therefore, we refer the interested reader to a book on convex optimization in general (Boyd and Vandenberghe 2004), and to an excellent recent review article on the application of semidefinite relaxations to signal-processing problems (Luo et al 2010).

In order to perform this semidefinite relaxation, observe that only quadratic forms of $\mathbf{q}$ occur in this problem. Let us define the matrix $\mathbf{T}$ of all the quadratic forms of $\mathbf{q}$, i.e.,

$$\mathbf{T} \equiv \mathbf{q}\mathbf{q}^T = \begin{bmatrix} r_0 \\ \mathbf{r} \end{bmatrix} \begin{bmatrix} r_0 & \mathbf{r}^T \end{bmatrix} \equiv \begin{bmatrix} s_0 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{bmatrix}. \tag{7}$$

From eq. (3), matrix $\mathbf{Q}$ can then be expressed as a linear function of $\mathbf{T}$. Problem (P) applied over the box $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$ is then rewritten in the equivalent form

$$\begin{aligned}
&\text{minimize} && V = -m\mathbf{g}^T\mathbf{p}, && \text{(8a)}\\
&\text{subject to} && \|\mathbf{c}_j\|_2 \le c_j,\ j = 1,\dots,n, && \text{(8b)}\\
& && \mathbf{c}_j = \mathbf{a}_j - \mathbf{Q}\mathbf{b}_j - \mathbf{p}, && \text{(8c)}\\
& && \mathbf{Q} = (s_0 - \mathrm{tr}(\mathbf{S}))\mathbf{1}_{3\times 3} + 2\mathbf{S} + 2\mathrm{cpm}(\mathbf{s}), && \text{(8d)}\\
& && 1 = s_0 + \mathrm{tr}(\mathbf{S}), && \text{(8e)}\\
& && \mathbf{q}\mathbf{q}^T = \mathbf{T} = \begin{bmatrix} s_0 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{bmatrix}, && \text{(8f)}\\
& && \underline{\mathbf{q}} \le \mathbf{q} \le \overline{\mathbf{q}}, && \text{(8g)}\\
&\text{over} && \mathbf{p} \in \mathbb{R}^3,\ \mathbf{q} \in \mathbb{R}^4.
\end{aligned}$$

This expression is useful in that it concentrates the nonconvex constraints into eq. (8f). This latter constraint is equivalent to the two constraints

$$\begin{bmatrix} 1 & \mathbf{q}^T \\ \mathbf{q} & \mathbf{T} \end{bmatrix} = \begin{bmatrix} 1 & r_0 & \mathbf{r}^T \\ r_0 & s_0 & \mathbf{s}^T \\ \mathbf{r} & \mathbf{s} & \mathbf{S} \end{bmatrix} \succeq 0, \tag{9a}$$

where $\succeq 0$ denotes nonnegative definiteness of its left-hand side matrix argument, and

$$\operatorname{rank}\left(\begin{bmatrix} 1 & r_0 & \mathbf{r}^T \\ r_0 & s_0 & \mathbf{s}^T \\ \mathbf{r} & \mathbf{s} & \mathbf{S} \end{bmatrix}\right) = 1. \tag{9b}$$

Indeed, the two constraints (9a) and (9b) are necessary and sufficient conditions for the existence of a real vector $\mathbf{q}$ satisfying eq. (8f) (see Luo et al 2010). As a result, constraint (8f) is equivalent to constraints (9a) and (9b).

The convex relaxation of problem (P) is obtained by dropping constraint (9b). This convex relaxation, however, is not in the form of any standard convex optimization program. We reformulate it as a semidefinite program (SDP) by noticing that the constraint (8b) is equivalent to

$$\begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{c}_j \\ \mathbf{c}_j^T & c_j^2 \end{bmatrix} \succeq 0. \tag{10}$$

For more details, the reader is referred to Vandenberghe and Boyd (1996).

Upon substituting eqs. (9a) and (10) for eqs. (8b) and (8f), we obtain

$$\begin{aligned}
&\text{minimize} && \underline{V} = -m\mathbf{g}^T\mathbf{p}, && \text{(SDR-1)}\\
&\text{subject to} && 0 \preceq \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{c}_j \\ \mathbf{c}_j^T & c_j^2 \end{bmatrix},\ j = 1,\ldots,n,\\
& && \mathbf{c}_j = \mathbf{a}_j - \mathbf{Q}\mathbf{b}_j - \mathbf{p},\\
& && \mathbf{Q} = (s_0 - \operatorname{tr}(\mathbf{S}))\mathbf{1}_{3\times 3} + 2\mathbf{S} + 2\operatorname{cpm}(\mathbf{s}),\\
& && 1 = s_0 + \operatorname{tr}(\mathbf{S}),\\
& && \mathbf{T} = \begin{bmatrix} s_0 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{bmatrix},\\
& && 0 \preceq \begin{bmatrix} 1 & \mathbf{q}^T \\ \mathbf{q} & \mathbf{T} \end{bmatrix},\\
& && \underline{\mathbf{q}} \le \mathbf{q} \le \overline{\mathbf{q}},\\
&\text{over} && \mathbf{p} \in \mathbb{R}^3,\ \mathbf{q} \in \mathbb{R}^4,\ \mathbf{T} \in \mathbb{R}^{4\times 4}.
\end{aligned}$$

Since all constraints are linear in the optimization variables, they are either linear equalities or linear matrix inequalities, and (SDR-1) is a semidefinite relaxation of (P). Hence, the optimum value of (SDR-1) may be computed using readily-available software, and represents a lower bound on the optimum value of (P).

Numerical experiments show that the lower bounds offered by this relaxation often result in a low rate of convergence of the corresponding branch-and-bound algorithm. Resolving this problem requires obtaining a lower bound that is more accurate,

i.e., a tighter approximation of problem (8). This can be done by adding convex constraints, which is done in the Appendix. Upon appending these constraints to problem (SDR-1), a final lower bound on problem (P) over the box $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$ is obtained as the solution of the convex semidefinite programming problem

$$
\begin{aligned}
\text{minimize} \quad & \underline{V} = -m\mathbf{g}^T\mathbf{p}, && \text{(SDR-2)}\\
\text{subject to} \quad & 0 \preceq \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{c}_j \\ \mathbf{c}_j^T & c_j^2 \end{bmatrix}, \ j = 1, \ldots, n,\\
& \mathbf{c}_j = \mathbf{a}_j - \mathbf{Q}\mathbf{b}_j - \mathbf{p},\\
& \mathbf{Q} = (s_0 - \mathrm{tr}(\mathbf{S}))\mathbf{1}_{3\times 3} + 2\mathbf{S} + 2\mathrm{cpm}(\mathbf{s}),\\
& 1 = s_0 + \mathrm{tr}(\mathbf{S}),\\
& \mathbf{T} = \begin{bmatrix} s_0 & \mathbf{s}^T \\ \mathbf{s} & \mathbf{S} \end{bmatrix},\\
& 0 \preceq \begin{bmatrix} 1 & \mathbf{q}^T \\ \mathbf{q} & \mathbf{T} \end{bmatrix},\\
& \underline{\mathbf{q}} \le \mathbf{q} \le \overline{\mathbf{q}},\\
& \mathbf{T} \le \underline{\mathbf{q}}\mathbf{q}^T + \mathbf{q}\overline{\mathbf{q}}^T - \underline{\mathbf{q}}\overline{\mathbf{q}}^T,\\
& \mathbf{T} \ge \mathbf{q}\underline{\mathbf{q}}^T + \underline{\mathbf{q}}\mathbf{q}^T - \underline{\mathbf{q}}\underline{\mathbf{q}}^T,\\
& \mathbf{T} \ge \mathbf{q}\overline{\mathbf{q}}^T + \overline{\mathbf{q}}\mathbf{q}^T - \overline{\mathbf{q}}\overline{\mathbf{q}}^T,\\
\text{over} \quad & \mathbf{p} \in \mathbb{R}^3, \ \mathbf{q} \in \mathbb{R}^4, \ \mathbf{T} \in \mathbb{R}^{4\times 4}.
\end{aligned}
$$

Letting $(\check{\mathbf{p}}, \check{\mathbf{q}}, \check{\mathbf{T}})$ be the solution to problem (SDR-2), and $(\mathbf{p}_*, \mathbf{q}_*)$ be the solution to problem (8), we obtain the inequality

$$\underline{V} \equiv -m\mathbf{g}^T\check{\mathbf{p}} \le -m\mathbf{g}^T\mathbf{p}_*. \tag{11}$$

3.3 Computing an Upper Bound

Although an upper bound to the global optimum of problem (P) over the box $\mathbf{q} \in [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$ may be obtained through any descent method, we adopt a different approach. The technique we use here is outlined by Luo et al (2010), and stems from the observation that a solution $(\check{\mathbf{p}}, \check{\mathbf{q}}, \check{\mathbf{T}})$ to problem (SDR-2) is the global optimum of problem (8) if

$$\mathrm{rank}\left(\begin{bmatrix} 1 & \check{\mathbf{q}}^T \\ \check{\mathbf{q}} & \check{\mathbf{T}} \end{bmatrix}\right) = 1. \tag{12}$$

When this last condition is not met, choosing the rank-one matrix that is *closest* to

$$\begin{bmatrix} 1 & \check{\mathbf{q}}^T \\ \check{\mathbf{q}} & \check{\mathbf{T}} \end{bmatrix} \tag{13}$$

generally yields a *good* estimate of the global optimum of the original problem.

When closeness is measured by means of the two-norm of the difference of the two matrices, the rank-one matrix closest to that of eq. (13) is

$$\lambda_1 \mathbf{u}_1 \mathbf{u}_1^T, \tag{14}$$

where λ_1 and $\mathbf{u}_1$ are the largest eigenvalue of matrix (13) and its corresponding eigenvector, respectively (Luo et al 2010). Therefore, a good estimate of $\mathbf{q}_*$ is $\sqrt{\lambda_1}\mathbf{u}_{1,1}$, where $\mathbf{u}_1 \equiv [u_{1,0} \quad \mathbf{u}_{1,1}^T]^T$. Taking into account the fact that $\mathbf{q}_*$ lies on the unit sphere and has a nonnegative first component, we normalize and flip the sign if necessary, which gives the better estimate $\hat{\mathbf{q}} = \mathrm{sgn}(u_{1,1})\mathbf{u}_{1,1}/\|\mathbf{u}_{1,1}\|_2$, where $u_{1,1}$ is the first component of $\mathbf{u}_{1,1}$. Having computed an estimate $\hat{\mathbf{q}}$ that lies on the unit-hypersphere of ER parameters, we can compute a valid proper-orthogonal rotation matrix $\hat{\mathbf{Q}}$ by substitution in eq. (3).

We cannot overlook the possibility that the estimate $\hat{\mathbf{q}}$ obtained falls outside of the considered box $[\underline{\mathbf{q}}, \overline{\mathbf{q}}]$. In such a case, we simply disregard the upper bound by setting its value to that of the upper bound of its parent box. Notice that the estimate $\hat{\mathbf{q}}$ of the initial box defined in eq. (6) is bound to fall inside this box, as, from its definition, $\hat{\mathbf{q}}$ lies on the top half of the unit hypersphere, which is completely contained in the initial box. Hence, the algorithm always has a valid upper bound to start from. Experience shows that simply keeping the upper bound of the parent box when $\hat{\mathbf{q}} \notin [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$ this approach does not prevent the algorithm from converging to the global optimum, since this situation does not occur as we approach $\mathbf{q}_*$.

Finally, if $\hat{\mathbf{q}} \in [\underline{\mathbf{q}}, \overline{\mathbf{q}}]$, we have a valid estimate of $\mathbf{q}$ in hand, which allows us to return to the unrelaxed problem (8). By setting $\mathbf{q} = \hat{\mathbf{q}}$, we are left with $\mathbf{p}$ as only optimization variable, turning the nonconvex problem into a convex one, or, more precisely, into a second-order cone program (SOCP). Such a problem is quickly solved using readily available algorithms, leading to an estimate $\hat{\mathbf{p}}$ of $\mathbf{p}_*$, the lowest position of point G. This estimate provides an upper bound $\overline{V}$ on the minimum potential energy, i.e.,

$$-m\mathbf{g}^T\mathbf{p}_* \leq -m\mathbf{g}^T\hat{\mathbf{p}} \equiv \overline{V}. \tag{15}$$

3.4 Summary

Figure 2 summarizes the proposed algorithm for computing the global minimum of problem (P), i.e., the lowest equilibrium pose of the cable-suspended rigid body. As detailed in section 3.1, we partition the four-dimensional space of Euler-Rodrigues parameters $\mathbf{q}$ using boxes. These boxes are stored in list $\mathscr{L}$, which initially contains only the box defined in eq. (6). Similarly, the lower and upper bounds on the potential energy over the corresponding boxes are stored in the lists $\underline{\mathscr{V}}$ and $\overline{\mathscr{V}}$, respectively. The box containing the lowest lower bound $\min_i \underline{V}_i$ is partitioned recursively, until the gap between the lower and upper bounds becomes smaller or equal to ε. Hence, ε represents the accuracy required on the potential energy of the suspended rigid body. Notice that the convergence of this algorithm in a finite number of steps is not demonstrated in this article, and it appears to be a daunting task. For instance, such a result

was obtained no earlier than in 1998 in the case of a rectangular branch-and-bound algorithm applied to the separable concave program (Shectman and Sahinidis 1998), a problem that has drawn a large share of attention since the 1960s. Nevertheless, one may notice that problems (SDR-2) and (8) become equivalent whenever $\overline{\mathbf{q}} = \underline{\mathbf{q}} = \hat{\mathbf{q}}$, so that we then have $\underline{V} = \overline{V}$. This shows that the lower and upper bounds become equal when the boxes become *infinitely small*, but does not provide any information as to the rate of convergence.

Before illustrating the method, let us make a few remarks about the algorithm flowchart of Fig. 2 and especially the ending statements symbolized by double frames. At the end of the branch-and-bound process, the algorithm may either find the global optimum or guarantee that no solution exists. The latter conclusion may be reached in two cases. If the relaxed problem (SDR-2) admits no solution in the initial box $\mathscr{B}_1$, neither will problem (P). This also happens if problem (SDR-2) is infeasible in every box of $\mathscr{L}$, i.e., the lowest lower bound is infinity,[2] as shown at the bottom of the flowhart. On the other hand, if problem (P) is feasible, we can exploit condition (12) to detect the global solution of (P) at the first iteration. Otherwise, the algorithm stops when the lowest lower and upper bounds lie within the prescribed tolerance ε. Finally, to improve the efficency in case of global infeasibility, a pruning test has been inserted at the end of the main loop. Its purpose is to remove boxes from $\mathscr{L}$ for which the lower bound is greater than the lowest upper bound, thus reducing the number of boxes to keep in memory. This also reduces the number of required evaluations before concluding to the infeasibility of problem (P).

4 Numerical Examples

Three examples of increasing complexities are presented in this Section. Firstly, two basic examples taken from the literature are solved. Secondly, the minimum height of the previous example is solved in a few iterations after modifying its dimensions randomly. Thirdly, a large system of 1000 cables is solved to show the robustness and efficiency of the method. All these examples are implemented in Matlab using `CVX`, a package for the formulation and solution of convex programs (Grant and Boyd 2010, 2008).

4.1 Basic Examples

The proposed branch-and-bound method is first applied to two simple academic examples: a planar one and a three-dimensional one. The numerical data of these two examples are respectively taken from Carricato and Merlet (2010) and Jiang and Kumar (2010).

[2] By convention, the minimum objective value of an infeasible problem is set to infinity.

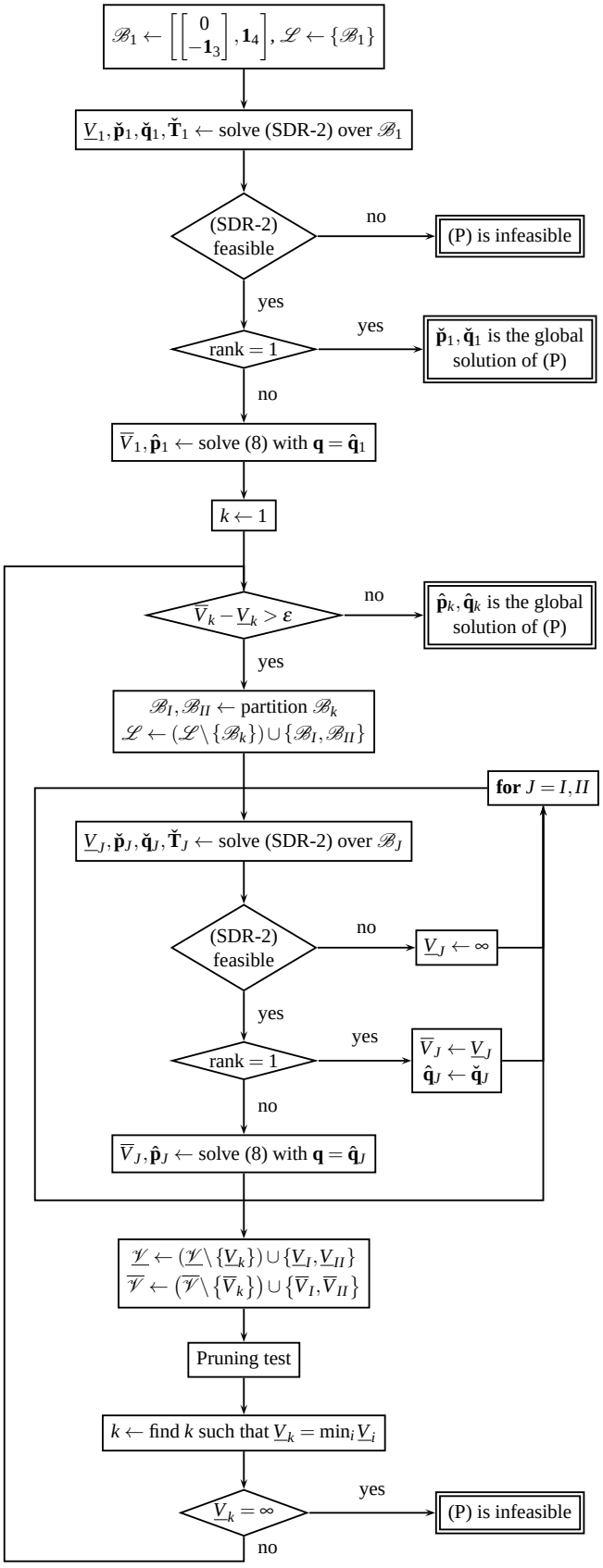

Fig. 2 Algorithm flowchart

4.1.1 Planar example (Carricato and Merlet 2010)

The rigid body is suspended by two cables of given lengths ($c_1 = c_2 = 7$) as illustrated in Fig. 3. In this first example, the solution of the relaxed problem (SDR-2) computed over the initial box gives a minimum height of -6.0714. The first eigenvector of matrix (13) is then used to produce an estimate $\hat{\mathbf{q}}_1$ of $\mathbf{q}_*$. Using this estimate to set $\mathbf{q} = \hat{\mathbf{q}}_1$ in problem (8), we obtain the same minimum height of -6.0714. Hence, the pose computed at the first iteration is the global optimum sought, which corresponds to the value given in (Carricato and Merlet 2010).

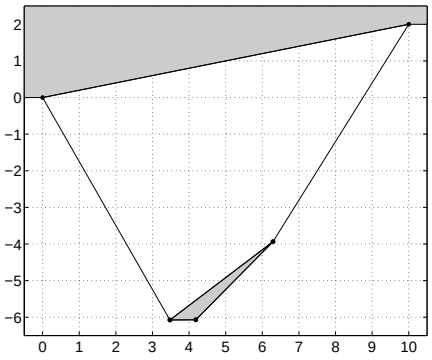

Fig. 3 A simple planar example from Carricato and Merlet (2010)

4.1.2 Three-dimensional example with equal cable lengths (Jiang and Kumar 2010)

Here, the object is suspended by six cables attached to six aerial robots. All cables have the same length ($c_j = 12$ m, $j = 1,\ldots,6$), the base and the platform are regular hexagons respectively of radii 4 m and 1 m, as depicted in Fig. 4. The center of mass of the platform is at its centroid. As for the previous example, the solution is straightforward. The relaxed problem (SDR-2) gives a minimum height of -11.6190 m. This is actually the global minimum, since the rank of matrix (13) is equal to one.

4.1.3 Three-dimensional example with different cable lengths (Jiang and Kumar 2010)

A second case is proposed where the cable lengths are different: $c_j = 8 + j$ m, $j = 1,\ldots,6$. The solution illustrated in Fig. 5 is still immediate, for the lower and upper bounds computed on the first box are both equal to -9.7556 m. In this particular case, it should be noted that only cables of lengths 9 m and 12 m are active. They are illustrated by thicker lines in Fig. 5, whereas the thinner lines do not represent the actual

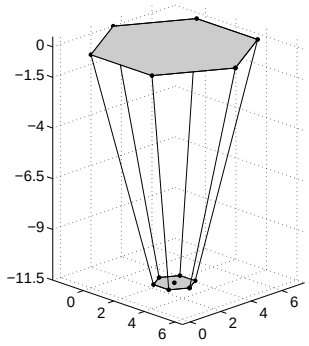

Fig. 4 A simple 3D example from Jiang and Kumar (2010)

lengths of the corresponding inactive sagging cables.

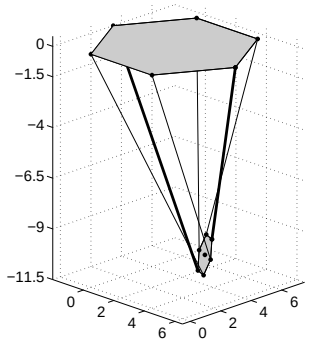

Fig. 5 A 3D example with different cable lengths from Jiang and Kumar (2010)

4.2 Irregular Six-Cable Example

In many regular situations, the relaxation of the problem proves to be tight and the global optimum is found after a single iteration. However, dealing with more irregular dimensions requires a few iterations for the proposed method to converge to the global minimum. In order to produce such an example, the dimensions of the previous six-cable example were randomly modified. The modifications of lengths c_j are within 10 % of their initial lengths (12 m), whereas the components of vectors $\mathbf{a}_j$ and $\mathbf{b}_j$ are modified within 10 % of the base and platform radii respectively ($j = 1, \ldots, 6$).

Several optimizations were carried out; All of them converged in less than a hundred iterations. One of these runs is illustrated in Fig. 7. In this graph, the values of the lowest lower and upper bounds come within less than 100 μm from one another in 41 iterations. Let us point out that the lowest upper bound does not decrease monotonically. Indeed, when partitioning a box in the branch-and-bound algorithm, the two replacing boxes may provide higher upper bounds since the estimate $\hat{\mathbf{q}}$ of $\mathbf{q}_*$ is then modified. The corresponding solution, which is represented in Fig. 6, has only three active cables.

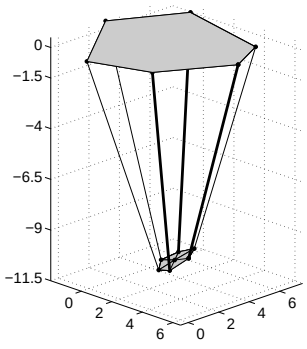

Fig. 6 An irregular 3D example derived from Jiang and Kumar (2010)

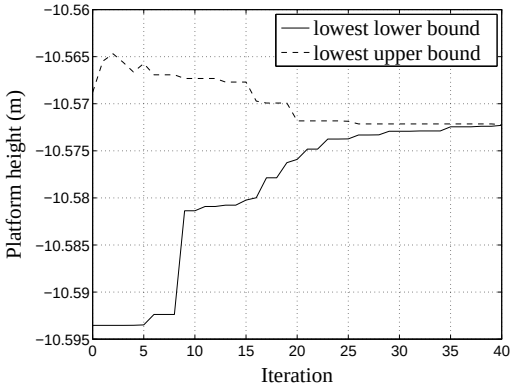

Fig. 7 Evolution of lowest lower and upper bounds in the proposed branch-and-bound algorithm to find the lowest equilibrium pose of the rigid body of example 2

4.3 Thousand-Cable Random Example

Not only the proposed branch-and-bound algorithm seems robust, but its computational complexity is polynomial with respect to the number of cables—the number of cable being proportional to the numbers of optimization variables and constraints. This interesting feature is now highlighted by seeking the minimum height of a rigid body suspended by 1000 cables. It must be emphasized that this example is far from the reality of cable-driven robots but could help in other fields of mechanics, e.g., in multi-contact modelling. The dimensions of this system were randomly generated in the interval $[10,20]$ for the cable lengths and the interval $[0,10]$ for the components of vectors $\mathbf{a}_j$ and $\mathbf{b}_j$, $j = 1,\ldots,1000$. As shown in Fig. 9, the process converges in 219 iterations to within a tolerance of 100 μm between the lowest lower and upper bounds. This takes 122 minutes with the `CVX` package in Matlab on a 2.50 GHz Centrino 2 processor. `CVX` rewrites problems (SDR-2) and (8) into dual problems comprising respectively 10095 and 4000 variables. This, again, illustrates the robustness and efficiency of the proposed method, which relies on convex optimization and the efficient algorithms that are available for such problems. The final solution is depicted in Fig. 8, where the three active cables are drawn in black and only a hundred inactive cables are represented in gray.

Finally, the computational performance of the proposed method over the given examples are summarized in Table 2. It can be observed that it generally takes less than one second to find the lowest equilibrium pose of a regular example or a few dozens of seconds if the dimensions are irregular. The computational time of a random large-scale example remains reasonnable regarding the size of the considered system.

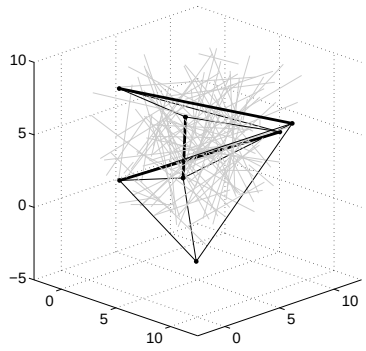

Fig. 8 Solution of the 1000-cable example: the three active cables are represented by thick black lines and 100 of the 997 inactive cables are drawn in gray

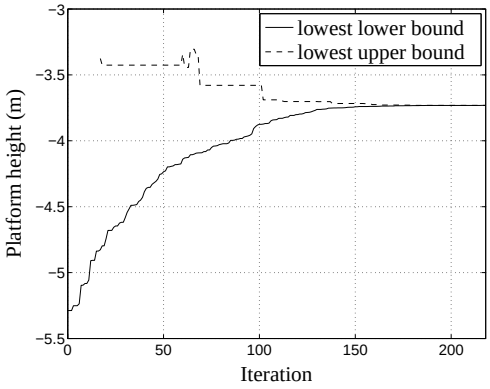

Fig. 9 Evolution of lowest lower and upper bounds in the proposed branch-and-bound algorithm to find the lowest equilibrium pose of a rigid body suspended by 1000 cables

Table 2 Computational Performance Obtained with the `CVX` Package in Matlab on a 2.50 GHz Centrino 2 Processor

Reference	Number of cables	Number of iterations	CPU time
4.1.1	2	1	< 1 s
4.1.2	6	1	< 1 s
4.1.3	6	1	< 1 s
4.2	6	41	32 s
4.3	1000	219	122 min

5 Conclusions

In summary, the proposed algorithm allows for the computation of the lowest equilibrium pose of *any* rigid-body suspended by n cables in space. The adopted branch-and-bound scheme reliably converges to the solution, and was even shown to sometimes yield the global optimum at the first iteration. Another interesting feature of the devised algorithm is the low sensitivity of its speed with respect to the number of cables involved in the problem, which was illustrated in Section 4.3 with the 1000-cable example.

We should also acknowledge some important drawbacks of the proposed method. Firstly, this approach does not provide any information regarding local equilibrium poses at which the rigid body may stabilize without reaching the lowest one. This stems from the branch-and-bound procedure, which abstracts local minima in its search for the global minimum. Secondly, we do not account for possible interference between two cables or between the rigid body and a cable. In view of the local-equilibrium and interference problems, the computed lowest equilibrium pose may be regarded as a lower bound on the height of the rigid body. Thirdly, a major challenge that remains to be addressed is that the proposed algorithm is not yet fast enough for real-time implementations. Indeed, the examples reported in Section 4 showed

running times in the orders of tens of seconds, whereas real-time control would require running times of at most tens of milliseconds. A viable direction may be the one pointed by Fink et al (2009). These authors provided geometric conditions under which the second-order cone relaxation of the optimization problem they propose is tight. Similarly, it may be possible to find classes of cable arrangements for which the semidefinite relaxation (SDR-2) is guaranteed to yield the global optimum.

Other future work will aim at extending the proposed method to find all the solutions of the forward kinematics problem, i.e. all the local optima of our optimization problem. This involves the modification of the branch-and-bound approach to explore the whole parameter space. Hence, despite being currently limited to the computation of the lowest equilibrium pose, we believe that the method proposed here provides a framework for the generalization to the computation of *all* equilibrium poses. This framework may be more viable than one based on algebraic geometry and polynomial continuation, as the complexity of such methods grows rapidly with the number of cables. This would explain why the latter techniques have been applied to rigid bodies suspended by three cables (Carricato and Merlet 2011) or less, for arbitrary cable attachment points and lengths. Another perspective of this research deals with accelerating the proposed method by refining its computer implementation and its contents. On the implementation side, translating the algorithm in a lower-level programming language such as C should result in an appreciable gain in speed. Bypassing `CVX`, which acts as an interpreter for convex optimization solvers, should also speed up the bounding steps. Regardless of the practical implementation, we believe that there remains room for improvement of the algorithm if one were to find a more efficient partitioning strategy for SO(3). For this purpose, other sets of parameters different from those of Euler-Rodrigues may lead to some improvement.

Acknowledgements The authors gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chair program as well as a FSR postdoctoral grant from the Université catholique de Louvain.

References

Angeles J (2007) Fundamentals of Robotic Mechanical Systems, 3rd edn. Springer, New York

Balakrishnan V, Boyd S, Balemi S (1991) Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. International Journal of Robust and Nonlinear Control 1(4):295–317

Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press, Cambridge, UK

Carricato M, Merlet JP (2010) Geometrico-static analysis of under-constrained cable-driven parallel robots. In: Lenarcic J, Stanisic M (eds) Advances in Robot Kinematics: Motion in Man and Machine, Springer, Dordrecht, pp 309–319

Carricato M, Merlet JP (2011) Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables. In: IEEE International Conference on Robotics and Automation, Shanghai, China, pp 3011–3017

Fink J, Michael N, Kim S, Kumar V (2009) Planning and control for cooperative manipulation and transportation with aerial robots. In: International Symposium on Robotics Research

Ghasemi A, Eghtesad M, Farid M (2010) Neural network solution for forward kinematics problem of cable robots. Journal of Intelligent & Robotic Systems 60(2):201–215

Grant M, Boyd S (2008) Graph implementations for nonsmooth convex programs. In: Blondel V, Boyd S, Kimura H (eds) Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, pp 95–110, http://stanford.edu/~boyd/graph_dcp.html
Grant M, Boyd S (2010) CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx
Husty M (1996) An algorithm for solving the direct kinematics of general Stewart-GOUGH platforms. Mechanism and Machine Theory 31(4):365–379
Jiang Q, Kumar V (2010) The direct kinematics of objects suspended from cables. In: ASME International Design Engineering Technical Conferences, Montreal, QC, Canada
Kawamura S, Choe W, Tanaka S, Pandian SR (1995) Development of an ultrahigh speed robot falcon using wire driven system. In: IEEE International Conference on Robotics and Automation, pp 215–220
Land AH, Doig AG (1960) An automatic method of solving discrete programming problems. Econometrica 28(3):497–520
Luo ZQ, Ma WK, So AMC, Ye Y, Zhang S (2010) Semidefinite relaxation of quadratic optimization problems. IEEE Signal Processing Magazine 27(3):20–34
Michael N, Kim S, Fink J, Kumar V (2009) Kinematics and statics of cooperative multi-robot aerial manipulation with cables. In: ASME International Design Engineering Technical Conferences, San Diego, CA
Perreault S, Gosselin C (2008) Cable-driven parallel mechanisms: Application to a locomotion interface. ASME Journal of Mechanical Design 130, article 102301 (8 pages)
Shectman JP, Sahinidis NV (1998) A finite algorithm for global minimization of separable concave programs. Journal of Global Optimization 12(1):1–35
Sherali HD, Tuncbilek CH (1992) A global optimization algorithm for polynomial problems using a reformulation-linearization technique. Journal of Global Optimization 2(1):101–112
Vandenberghe L, Boyd S (1996) Semidefinite programming. SIAM Review 38(1):49–95
Wampler CW (1996) Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. Mechanism and Machine Theory 31(3):331–337

Appendix

The constraints required to tighten the relaxation (SDR-1) are obtained from the "reformulation-linearization technique" (Sherali and Tuncbilek 1992). First consider the outer products

$$\begin{aligned}(\mathbf{q}-\underline{\mathbf{q}})(\overline{\mathbf{q}}-\mathbf{q})^T &= -\mathbf{q}\mathbf{q}^T+\underline{\mathbf{q}}\mathbf{q}^T+\mathbf{q}\overline{\mathbf{q}}^T-\underline{\mathbf{q}}\overline{\mathbf{q}}^T \geq \mathbf{0}_{4\times 4},\\ (\mathbf{q}-\underline{\mathbf{q}})(\mathbf{q}-\underline{\mathbf{q}})^T &= \mathbf{q}\mathbf{q}^T-\mathbf{q}\underline{\mathbf{q}}^T-\underline{\mathbf{q}}\mathbf{q}^T+\underline{\mathbf{q}}\underline{\mathbf{q}}^T \geq \mathbf{0}_{4\times 4},\\ (\overline{\mathbf{q}}-\mathbf{q})(\overline{\mathbf{q}}-\mathbf{q})^T &= \mathbf{q}\mathbf{q}^T-\mathbf{q}\overline{\mathbf{q}}^T-\overline{\mathbf{q}}\mathbf{q}^T+\overline{\mathbf{q}}\overline{\mathbf{q}}^T \geq \mathbf{0}_{4\times 4},\end{aligned} \tag{16}$$

where, in this case, $\geq$ denotes the componentwise inequality between the left- and right-hand-side matrices. Upon substituting eq. (7) into eqs. (16), we obtain the additional (convex) linear inequalities appearing in problem (SDR-2), namely,

$$\begin{aligned}-\mathbf{T}+\underline{\mathbf{q}}\mathbf{q}^T+\mathbf{q}\overline{\mathbf{q}}^T-\underline{\mathbf{q}}\overline{\mathbf{q}}^T &\geq \mathbf{0}_{4\times 4},\\ \mathbf{T}-\mathbf{q}\underline{\mathbf{q}}^T-\underline{\mathbf{q}}\mathbf{q}^T+\underline{\mathbf{q}}\underline{\mathbf{q}}^T &\geq \mathbf{0}_{4\times 4},\\ \mathbf{T}-\mathbf{q}\overline{\mathbf{q}}^T-\overline{\mathbf{q}}\mathbf{q}^T+\overline{\mathbf{q}}\overline{\mathbf{q}}^T &\geq \mathbf{0}_{4\times 4}.\end{aligned} \tag{17}$$