



ARCHIVIO ISTITUZIONALE DELLA RICERCA

Alma Mater Studiorum Università di Bologna Archivio istituzionale della ricerca

Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis / Berti, A.; Merlet, J.-P.; Carricato, M.. - In: THE INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH. - ISSN 0278-3649. - STAMPA. - 35:6(2016), pp. 723-739. [10.1177/0278364915595277]

This version is available at: <https://hdl.handle.net/11585/515621> since: 2020-01-31

Published:

DOI: <http://doi.org/10.1177/0278364915595277>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

(Article begins on next page)

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

This is the final peer-reviewed accepted manuscript of:

Alessandro Berti, Jean-Pierre Merlet and Marco Carricato

“Solving the direct geometrico-static problem of underconstrained cable-driven parallel robots by interval analysis”

The International Journal of Robotics Research, 2016, vol. 35(6) 723–739

Copyright © The Authors 2015

DOI: 10.1177/0278364915595277

Solving the Direct Geometrico-Static Problem of Underconstrained Cable-Driven Parallel Robots by Interval Analysis

The International Journal of Robotics
Research
000(00):1–26
©The Author(s) 2010
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI:doi number
<http://mms.sagepub.com>

Alessandro Berti

Department of Industrial Engineering (DIN), University of Bologna, IT

Jean-Pierre Merlet

National Research Institute in Computer Science and Automation (INRIA), Sophia Antipolis, FR

Marco Carricato

Department of Industrial Engineering (DIN), University of Bologna, IT

Abstract

This paper presents an efficient interval-analysis-based algorithm to solve the direct geometrico-static problem (DGP) of underconstrained cable-driven parallel robots (CDPRs). Solving the DGP for a generic CDPR consists in finding all possible equilibrium poses of the end-effector for given cable lengths. Since cables impose unilateral constraints, configurations with one or more slack cables may occur. When the number of taut cables is smaller than 6, the robot is underconstrained, and DGP solutions must be found considering loop-closure and mechanical equilibrium equations simultaneously. The presented algorithm can find all DGP solutions of a generic CDPR in a numerically robust and safe way. By using interval analysis, the proposed procedure can directly search for real solutions with nonnegative cable tensions, and it can take advantage of the physical constraints of the robot. The implemented procedure is discussed in detail, and testing and experimental validation on working prototypes are presented.

Keywords

Cable-driven parallel robots (CDPRs), underconstrained robots, interval analysis, forward kinematics, static analysis.

1. Introduction

Cable-driven parallel robots (CDPRs) control the end-effector pose using cables instead of rigid-body extensible legs. CDPRs enhance classic advantages characterizing parallel manipulators versus serial ones, like reduced mass and inertia, a larger payload to robot weight ratio, high dynamic performances, etc., and provide a larger workspace, reduced manufacturing and maintenance costs, high transportability, superior modularity and reconfigurability.

Since cables can exert forces only when they are taut, additional actuators (with respect to the number of freedoms to be controlled) can be used to prevent cables from becoming slack. This is often the case for redundantly-actuated cable-driven robots (Bosscher et al. 2007, Hiller et al. 2005, Kawamura et al. 2000, Kossowski and Notash 2002, Ming and Higuchi 1994b, Tadokoro et al. 2002). This redundancy can be reduced if a constant external force (with convenient magnitude and direction) acts on the end-effector. For example, this happens when CDPRs are in crane configuration, where gravity acts like an additional cable (Albus et al. 1993, Su et al. 2001). During the past decades, this kind of robot, usually called *fully-constrained* or *completely-restrained*, attracted increasing interest among researchers (Albus et al. 1993, Behzadipour and Khajepour 2006, Bosscher et al. 2006, 2007, Bouchard et al. 2010, Gouttefarde et al. 2011, Hiller et al. 2005, Kawamura et al. 2000, Kossowski and Notash 2002, Kozak et al. 2006, Merlet 2004a, Ming and Higuchi 1994a,b, Pham et al. 2009, Pott et al. 2013, Pusey et al. 2004, Roberts et al. 1998, Stump and Kumar 2006, Su et al. 2001, Tadokoro et al. 2002), and several applications were developed or are currently investigated (displacement of materials over large areas, positioning systems for heavy objects, pick-and-place, rescue operations, service, assistance, rehabilitation, entertainment, etc.).

This paper focuses, instead, on *underconstrained* CDPRs, i.e. robots either having a number n of cables smaller than 6 or, if $n \geq 6$, operating as such because the external wrench applied to the end-effector would require a negative tension in more than $n - 6$ cables. The first reason that encourages a careful study of underconstrained cable robots is that they offer peculiar advantages with respect to fully-constrained ones. A smaller number of cables, indeed, means reduced costs and setup time, improved ease of assembly and lower likelihood of cable interference. These features make underconstrained CDPRs particularly well-suited for all those applications in which a limited dexterity is acceptable in order to decrease complexity, such as rehabilitation tasks, rescue operations, positioning of objects over large workspaces, etc. (Merlet and Daney 2010, Rosati et al. 2007, Surdilovic et al. 2007, Tadokoro et al. 1999). With respect to fully-constrained robots, little attention has been dedicated to underconstrained CDPRs (Collard and Cardou 2013, Fattah and Agrawal 2005, Fink et al. 2011, Gao et al. 2012, Heyden and Woernle 2006, Jiang and Kumar 2010, Yamamoto et al. 2004).

The most challenging aspect in the kinematic study of underconstrained CDPRs comes from the fact that only less than 6 dof may be controlled, so that when the cable lengths are assigned the end-effector still has some freedoms. Thus, its actual pose is determined by the wrenches acting upon it. As a consequence, loop-closure and mechanical-equilibrium equations must be simultaneously solved, and traditional displacement-analysis problems become, in fact, *geometrico-static*. Moreover, as the pose of the end-effector is determined by the applied forces, it may change due to external disturbances (e.g. a wind blast), so considering equilibrium stability is necessary.

This paper solves the *direct geometrico-static problem* (DGP) of a generic n - n CDPR ($n \leq 6$), namely a robot in which the n exit points of cables on the base and the n cable anchor points on the platform are distinct. The DGP consists in assigning the cable lengths as input variables and in finding *all* possible equilibrium configurations, taking into account the fact that one or more cables may be slack. While the DGP of a CDPR having 6 taut cables can be dealt with by means of the same tools used for the Gough-Stewart Platform, the DGP for robots with less than 6 cables in tension is remarkably more complex. A successful implementation of the methodology proposed by Carricato and Merlet (Carricato and Merlet 2010, 2013), based on exact-arithmetic elimination procedures, allowed the DGP of 2-2, 3-3, 4-4 and 5-5 CDPRs to be solved (Abbasnejad and Carricato 2015, Carricato 2013, Carricato and Merlet 2011, 2013). In particular, least-degree univariate polynomials in the ideals generated by the equations governing the problems were found. However, the approach used in the aforementioned contributions has the following drawbacks.

- Obtaining univariate polynomials is a challenging and not automatic process. Moreover, they have high order (12, 156, 216 and 140, for the robots suspended by 2, 3, 4 and 5 taut cables, respectively), so that their reliable solution is very difficult, as the calculation of coefficients is very sensitive to numerical errors.
- It is not possible to incorporate constraints on the unknowns. As a consequence, all roots (both complex and real, regardless of tension sign and stability) must be calculated and then post-processed in order to discard unfeasible ones.

Effective alternatives are provided by approaches based on floating-point arithmetic, such as homotopy continuation or interval analysis. Continuation methods proved to be able to robustly provide all solutions for the DGP of a general n - n CDPR (Abbasnejad and Carricato 2012, 2015). However, they may be slow, and it is impossible to incorporate constraints on the unknowns.

Interval-analysis-based methods may be very effective in robotic applications (Merlet 2004b, 2009), and a specific procedure for the DGP of a 3-3 CDPR was recently developed with promising results (Berti et al. 2013). Based on these results, this paper extends and improves the algorithm presented in (Berti et al. 2013), thus making it suitable to solve the DGP of a general CDPR suspended by n cables, with $n = 1 \dots 6$, under the assumption that cables are massless and perfectly stiff. The same algorithm can also be used to solve the DGP of robots suspended by more than 6 cables, when cable elasticity is disregarded (Merlet 2012).

The paper is organized as follows. Section 2 provides basic notions of interval analysis and outlines the general scheme of the problem-solving algorithm. Section 3 presents the robot model. Section 4 describes the structure of the most relevant parts of the code and the procedures incorporated therein. Section 5 presents the results obtained from case studies and gives some insights on the performances of the code and its behavior. Section 6 draws some conclusions.

2. Interval analysis

2.1. Basic notions

Interval analysis is a numerical technique that allows all real solutions of a system of equations and/or inequalities to be obtained in a predetermined search domain (Hansen and Walster 2003, Jaulin 2001, Moore 1966). The *real interval* $X = [\underline{x}, \bar{x}]$ is defined as the set of real numbers y such that $\underline{x} \leq y \leq \bar{x}$. The *width* and the *mid-point* of the interval are given, respectively, by functions $w(X) = \bar{x} - \underline{x}$ and $\text{mid}(X) = (\underline{x} + \bar{x})/2$. An *interval vector* \mathbf{B} , also called a *box*, is a list of intervals. The *mid-point* and the *width* of a box are vectors whose components are respectively the mid-points and the widths of their interval components.

Let $f(\mathbf{x})$ be a function in h unknowns, i.e. $\mathbf{x} = [x_1, x_2, \dots, x_h]$, and $\mathbf{B} = [X_1, X_2, \dots, X_h]$ a box comprising an interval for each unknown. The *interval evaluation* $F(\mathbf{B})$ of f over \mathbf{B} is an interval $[\underline{F}, \bar{F}]$ such that, for any $\mathbf{x} \in \mathbf{B}$, $\underline{F} \leq f(\mathbf{x}) \leq \bar{F}$, namely \underline{F} and \bar{F} are lower and upper bounds for f when the unknowns lie in \mathbf{B} .

Interval analysis provides several tools to implement an interval evaluation of a function (Moore 1979), but the simplest one is the *natural evaluation*, in which basic arithmetic operations and elementary mathematical functions are substituted by interval equivalents. For example, if $f(x) = x^2 - 2x + 1$ and $X = [4, 5]$, the natural evaluation of f over X is:

$$f([4, 5]) = [4, 5]^2 - 2[4, 5] + 1 = [16, 25] - [8, 10] + [1, 1] = [6, 17] + [1, 1] = [7, 18] \quad (1)$$

The bounds provided by the natural evaluation of f are generally not exact: the upper (lower) bound may be larger (lower) than the actual maximum (minimum) of the function image, namely $f(\mathbf{B}) = \{f(\mathbf{x}) | \mathbf{x} \in \mathbf{B}\} \subseteq F(\mathbf{B})$. For example, the image of the function f in (1) over the interval $[4, 5]$ is $[9, 16] \in [7, 18]$. Such an overestimation dramatically affects the performances of interval-analysis methods. This is the reason why methods that provide the tightest interval evaluation of functions are of fundamental importance.

An interesting property is that the bounds of the interval evaluation F are exactly the minimum and the maximum of $f(\mathbf{B})$ when f may be expressed so as to contain a single occurrence of each unknown x_i , $i = 1, \dots, h$ (Jaulin 2001). Due to this, the expression in (1) may be more effectively evaluated as

$$f(X) = X^2 - 2X + 1 = (X - 1)^2 = [9, 16] \quad (2)$$

which are the sharpest bounds for f over $X = [4, 5]$. The aforementioned property infers that, when dealing with interval-analysis methods, the choice of the parameterization is extremely important. Indeed, if the problem is modeled by a set of parameters that lead to expressions having a large number of multiple occurrences, the interval evaluation of each function is heavily overestimated, and thus a large number of bisections may be necessary to assess whether a box may contain a solution or not (see Section 2.2). On the other hand, using a redundant parameterization to avoid multiple occurrences means a greater number of functions to be evaluated at each step and a larger number of bisections to be made to meet the stopping criterion of the algorithm for a possible solution box. Nonetheless, as a rule of thumb of interval-analysis-based methods, it is usually better to consider simpler expressions, even at a price of a larger number of unknowns, in order to avoid overestimation on the function evaluation.

The following properties also hold:

- if $0 \notin [\underline{F}, \overline{F}]$, then there is no value of \mathbf{x} such that $f(\mathbf{x}) = 0$;
- interval evaluation may be implemented on a computer in a ‘guaranteed’ way, by taking into account numerical round-off errors;
- interval arithmetic is not restricted to algebraic functions.

2.2. Problem-solving algorithm

The advantages provided by interval methods for the solution of systems of nonlinear equations are multiple (Merlet 2009). One of the most important is the capability to obtain results guaranteed against numerical errors. Moreover, interval-analysis algorithms have the advantage of directly searching for real solutions, which are the only ones of practical interest. In addition, interval analysis searches for solutions only within a predetermined domain. This property turns out to be extremely useful in robotic applications, and in particular when dealing with kinematics problems. Indeed, for any kind of robot, the solutions of the forward kinematics have to lie within the specific workspace of the robot or, when inverse kinematics is considered, within the ranges of motion of its actuators. In the particular case of CDPRs, by taking advantage of this peculiar feature, only those solutions with positive cable tensions can be obtained (see Section 4).

The simplest problem-solving algorithm that can be implemented in interval arithmetic relies on the branch-and-bound (or branch-and-prune) scheme. However, this approach leads to unsatisfactory performances, unless a large number of tools and techniques are integrated to make the algorithm effective. This fact represents one of the main drawbacks of interval analysis, since choosing the right heuristics and the best way to implement them has a dramatic impact on the code performances, and it requires a lot of experience, whereas the literature devoted to the applications of interval analysis in robotics is limited.

The structure of the scheme used in our code to solve a system of h equations in h unknowns is described in Algorithm 1. Let $\mathbf{B}_1 = [X_1, X_2, \dots, X_h]$ be a box, and $\mathbf{f} = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_h(\mathbf{x})] = \mathbf{0}$ a vector equation to be solved within \mathbf{B}_1 . \mathcal{L} is a list of boxes, initially set as $\mathcal{L} = \{\mathbf{B}_1\}$. An index i , initialized to 1, indicates which box \mathbf{B}_i in \mathcal{L} is currently being processed, while N denotes the number of boxes in \mathcal{L} . \mathcal{S} is another list, initially empty, storing the solutions. The interval evaluation of f_j over \mathbf{B}_i is denoted as $F_j(\mathbf{B}_i)$, with $j = 1, \dots, h$.

A key component of the algorithm is the evaluation operator \mathcal{E} , which takes a box \mathbf{B}_i as an input, and returns:

- 1, if, for any j , both $w(F_j(\mathbf{B}_i))$ is smaller than a given threshold ε and $F_j(\mathbf{B}_i)$ includes 0; in this case, \mathbf{B}_i is deemed to be a solution, and it is stored in \mathcal{S} ;
- -1, if $F_j(\mathbf{B}_i)$ does not include 0 for at least one j ;
- 0, otherwise.

Algorithm 1 Problem-solving algorithm scheme

```

1:  $i = 1, N = 1, \mathcal{L} = \{\mathbf{B}_1\}, \mathcal{S} = \{\}$ 
2: if  $i > N$  then return  $\mathcal{S}$ ;
3: end if
4: if  $\mathcal{F}(\mathbf{B}_i) = -1$  then
5:    $i = i + 1$ 
6:   go to 2
7: else
8:    $\mathbf{B}_i = \mathcal{F}(\mathbf{B}_i)$ 
9: end if
10: compute  $\mathcal{E}(\mathbf{B}_i)$ 
11: if  $\mathcal{E}(\mathbf{B}_i) = -1$  then
12:    $i = i + 1$ 
13:   go to 2
14: end if
15: if  $\mathcal{E}(\mathbf{B}_i) = 1$  then
16:   add  $\mathbf{B}_i$  to  $\mathcal{S}$ 
17:    $i = i + 1$ 
18:   go to 2
19: end if
20: if  $\mathcal{E}(\mathbf{B}_i) = 0$  then
21:   pick  $x_k$  such that  $w(X_k) = \max(w(\mathbf{B}_i))$ 
22:   bisect  $X_k$  in the middle point
23:   create two new boxes  $\mathbf{B}'_i$  and  $\mathbf{B}''_i$  from  $\mathbf{B}_i$ 
24:   replace  $\mathbf{B}_i$  with  $\{\mathbf{B}'_i, \mathbf{B}''_i\}$  in  $\mathcal{L}$ 
25:    $N = N + 1, i = i + 1$ 
26:   go to 2
27: end if

```

Another key component is the filter operator \mathcal{F} , which takes a box as an input and returns:

- -1 , if there is no solution in the input box;
- a box smaller than the input one, if the filter determines that the removed part of the input box cannot contain a solution;
- the unchanged input box, otherwise.

A box which is neither discarded nor deemed to be a solution is bisected. The strategy adopted in the code to select which variable has to be bisected consists in picking the variable contained in the box \mathbf{B}_i having the largest width. A strength of an algorithm of this kind is that it always terminates, since the size of a box always decreases after a bisection. Furthermore, provided that the new boxes emerging from a bisection are put at the top of the list, there is usually no problem of memory storage. From tests conducted so far, since the initial domain of the search space is bounded by the robot physical dimensions (see Section 4) and the filter operator \mathcal{F} can often reduce the width of a box without bisecting it, the number of boxes processed by the algorithm with $\varepsilon = 10^{-9}$ is usually less than 15 thousands and list \mathcal{L} does not contain more than 40 boxes.

3. Equation sets for the direct geometrico-static analyses

In CDPRs, the unilateral nature of the constraints imposed by cables does not allow one to guarantee that in every configuration all cables remain taut. As a consequence, in order to obtain the complete set of possible solutions of the problem, all configurations with one or more slack cables must be considered. Accordingly, for a generic n - n CDPR, a complete solution of its DGP comprises not only the equilibrium configurations with n cables in tension, but also those with $1, 2, \dots, n-1$ cables being slack. Hence, the number of DGPs that must be solved is

$$N_{DGP} = \sum_{k=1}^n \frac{n!}{k!(n-k)!} \quad (3)$$

This section describes how the DGP is modeled for robots with m cables in tension, $m \leq n$, and which parameterizations and equation sets are used for each value of m . In the following, an equilibrium configuration is defined *admissible* if all cable tensions are positive or zero, and *feasible* if it is also stable.

3.1. Geometrico-static model

A mobile platform is connected to a fixed base by n cables. The i th cable, $i = 1, \dots, n$, exits from the base at point A_i , and it is connected to the platform at point B_i (Fig. 1). The platform is acted upon by a force of constant magnitude Q applied at point G , e.g. the platform weight acting through its center of mass. This force is described as a 0-pitch wrench $Q\mathcal{L}_e$, where \mathcal{L}_e is the normalized Plücker vector of its line of action. $Oxyz$ is a Cartesian coordinate frame fixed to the base, and $O'x'y'z'$ is a Cartesian frame attached to the end-effector (vector components in $O'x'y'z'$ are denoted with the prime mark). Without loss of generality, the fixed reference frame is chosen in such a way that $O \equiv A_1, A_2$ lies on the xz plane and z is directed as \mathcal{L}_e . Moreover: $\mathbf{a}_i = A_i - O = [a_{ix}, a_{iy}, a_{iz}]$, $\mathbf{b}_i = B_i - O = [x_i, y_i, z_i]$, $\mathbf{g} = G - O = [g_x, g_y, g_z]$, $\mathbf{b}'_i = B_i - O' = [x'_i, y'_i, z'_i]$, $\mathbf{g}' = G - O' = [g'_x, g'_y, g'_z]$, $d_{iG} = \|\mathbf{b}'_i - \mathbf{g}'\|$, and $d_{ij} = \|\mathbf{b}'_i - \mathbf{b}'_j\|$.

If ρ_i is the assigned length of the i th cable (taken as strictly positive), when m cables are *active* (i.e. in tension) the set of geometrical constraints imposed on the platform is formed by the following relations:

$$\|\mathbf{b}_i - \mathbf{a}_i\| = \rho_i, \quad i = 1, \dots, m \quad (4)$$

Since the platform has 6 dofs, when $m < 6$ its pose is ultimately determined by mechanical equilibrium. The normalized Plücker vector of the line associated with the i th cable is \mathcal{L}_i/ρ_i , where $\mathcal{L}_i = \begin{bmatrix} (\mathbf{a}_i - \mathbf{b}_i); & \mathbf{p}_i \times (\mathbf{a}_i - \mathbf{b}_i) \end{bmatrix}$ and \mathbf{p}_i is any

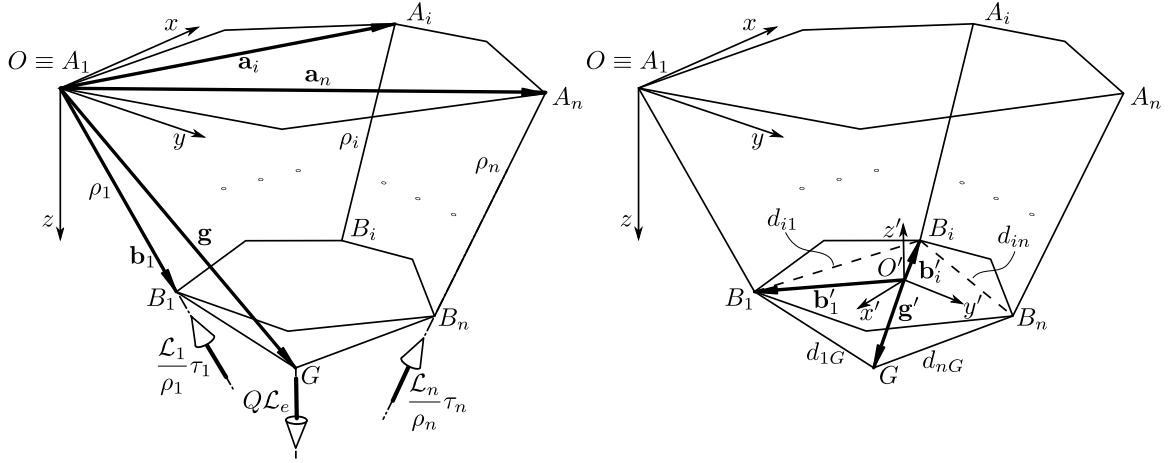


Fig. 1. The geometric model of a cable driven parallel robot with n cables.

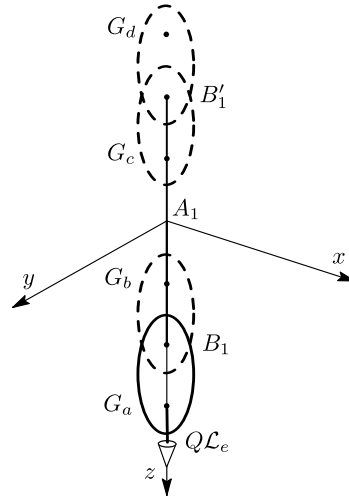


Fig. 2. The 4 possible solutions of the DGP with one active cable.

vector from an arbitrarily-chosen reference point P (called, for brevity, *moment pole*) to the cable line. Accordingly, the wrench exerted by the i th cable on the platform is $(\tau_i/\rho_i)\mathcal{L}_i$, with τ_i being a positive scalar representing the intensity of the cable tensile force. Static equilibrium may then be expressed as

$$\underbrace{\begin{bmatrix} \mathcal{L}_1 & \dots & \mathcal{L}_m & \mathcal{L}_e \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} \tau_1/\rho_1 \\ \vdots \\ \tau_m/\rho_m \\ Q \end{bmatrix} = \mathbf{0}, \quad (5)$$

with $\tau_i \geq 0$, $i = 1, \dots, m$.

3.2. DGP for robots with one cable in tension

When the platform is suspended by a single cable, under the effect of a force $Q\mathcal{L}_e$, it has 4 equilibrium configurations (Fig. 2), but only in two of them, i.e. G_a and G_b , is cable tension positive. Obviously, the only stable solution is the one having the minimum potential energy, i.e. G_a .



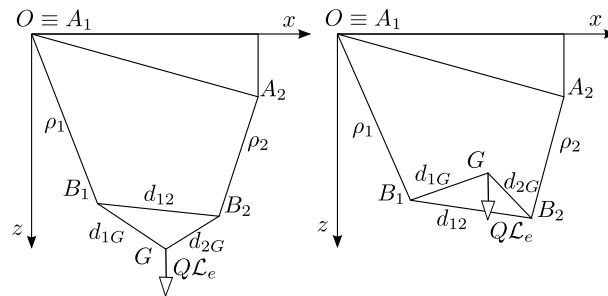
If the taut cables are labeled with 1 and 2, then, at equilibrium, A_1 , B_1 , A_2 , B_2 and G must necessarily rest in a plane parallel to the z axis. Two planar operation modes are possible, characterized by opposite orientation of the vector

Algorithm 2 DGP 1-1 for the i th cable in tension

```

1: for ( $j = 1 \dots n$ ) do
2:   if ( $j \neq i$ ) then
3:     if ( $a_{jx} = a_{ix}$ ) and ( $a_{jy} = a_{iy}$ ) then
4:        $D_{j\min} = r_{ij}^2 + (a_{iz} + \rho_i + z_{ij} - a_{jz})^2$ 
5:     else
6:       if ( $a_{jx} = a_{ix}$ ) then
7:         if ( $a_{jy} > a_{iy}$ ) then
8:            $\vartheta_{\min} = \pi/2$ 
9:         else
10:           $\vartheta_{\min} = 3\pi/2$ 
11:        end if
12:      else
13:        if ( $a_{jx} > a_{ix}$ ) then
14:           $\vartheta_{\min} = \arctan\left(\frac{a_{jy}-a_{iy}}{a_{jx}-a_{ix}}\right)$ 
15:        else
16:           $\vartheta_{\min} = \arctan\left(\frac{a_{jy}-a_{iy}}{a_{jx}-a_{ix}}\right) + \pi$ 
17:        end if
18:      end if
19:       $D_{j\min} = D_j(\vartheta_{\min})$ 
20:    end if
21:    if ( $D_{j\min} > \rho_j$ ) then
22:      The configuration is unfeasible
23:      Exit
24:    end if
25:  end if
26: end for
27: The configuration is feasible
28: Exit

```

**Fig. 4.** Geometric model of the DGP with two cables in tension.

$(\mathbf{b}_2 - \mathbf{g}) \times (\mathbf{b}_1 - \mathbf{g})$ with respect to $\mathbf{k} \times (\mathbf{a}_2 - \mathbf{a}_1)$ (Fig. 4). Out-of-the-plane movements, which are very likely to occur in practice, can cause operation-mode changes. The platform pose is described by the 9 components of the position vectors \mathbf{b}_1 , \mathbf{b}_2 and \mathbf{g} . By conveniently rotating the fixed reference frame along the z axis in order to have the exit points A_1 and A_2 lying on the xz plane, the number of variables can be reduced to 6, since, at the equilibrium, the y component of \mathbf{b}_1 , \mathbf{b}_2 and \mathbf{g} must necessarily vanish. These 6 parameters are not independent, since they have to satisfy the distance relations imposed by the geometry of the platform, namely:

$$\begin{aligned} \|\mathbf{b}_2 - \mathbf{b}_1\| &= d_{12} \\ \|\mathbf{b}_1 - \mathbf{g}\| &= d_{1G} \\ \|\mathbf{b}_2 - \mathbf{g}\| &= d_{2G} \end{aligned} \quad (10)$$

By the aforementioned parameterization, (4) assumes the form

$$(x_i - a_{ix})^2 + (z_i - a_{iz})^2 = \rho_i^2, \quad i = 1, 2 \quad (11)$$

and, by choosing O as the moment pole, matrix \mathbf{M} in (5) may be written as

$$\mathbf{M} = \begin{bmatrix} x_1 & x_2 - a_{2x} & 0 \\ z_1 & z_2 - a_{2z} & -1 \\ 0 & a_{2z}x_2 - a_{2x}z_2 & x \end{bmatrix} \quad (12)$$

Substituting (12) into (5) and considering relations (10) and (11), a system of 8 equations in the 8 variables comprised in the array $\mathbf{Y} = [\mathbf{X}^T, \boldsymbol{\tau}^T]^T$ is finally obtained, where $\mathbf{X} = [x_1, z_1, x_2, z_2, x_g, z_g]^T$ and $\boldsymbol{\tau} = [\tau_1, \tau_2]^T$. This parameterization allows one to simultaneously search for solutions in both operation modes.

The number of unknowns (and consequently the number of equations) could be reduced by observing that the system in (5) admits a solution only if $\text{rank}(\mathbf{M}) \leq 2$. Thus, cable tensions $\boldsymbol{\tau}$ could be eliminated from equilibrium equations by imposing $\det(\mathbf{M}) = 0$. This strategy, applied in (Carricato and Merlet 2013), is not adopted here, because it does not allow an initial search domain for the cable tensions to be specified. In this case, negative-tension solutions should be filtered by a subsequent procedure that would increase computation time.

3.4. DGP for robots with three cables in tension

The geometrico-static analysis of this family of robots was investigated in (Abbasnejad and Carricato 2012, Carricato 2013, Carricato and Merlet 2011), where the DGP of a robot suspended by 3 active cables was proven to admit 156 solutions in the complex field. Preliminary results obtained by solving the same problem by interval analysis were presented in (Berti et al. 2013).

If the taut cables are labeled with 1, 2 and 3, the parameters \mathbf{X} adopted to describe the platform pose are the components of the position vectors \mathbf{b}_1 , \mathbf{b}_2 and \mathbf{b}_3 . They are not independent, since the following relations must hold:

$$\|\mathbf{b}_i - \mathbf{b}_j\| = d_{ij}, \quad i, j \in [1, 2, 3], \quad i \neq j \quad (13)$$

The distance equations in (4) assume the form

$$(x_i - a_{ix})^2 + (y_i - a_{iy})^2 + (z_i - a_{iz})^2 = \rho_i^2, \quad i = 1, 2, 3 \quad (14)$$

and the position vector of $(G - B_1)$ in $Oxyz$ may be expressed as

$$\mathbf{g} - \mathbf{b}_1 = \alpha_2 \mathbf{b}_{12} + \alpha_3 \mathbf{b}_{13} + \alpha_4 (\mathbf{b}_{12} \times \mathbf{b}_{13}) \quad (15)$$

where $\mathbf{b}_{ij} = \mathbf{b}_j - \mathbf{b}_i$, and $\alpha_k, k = 2, \dots, 4$, are known constants obtained by solving the system

$$\alpha_2 \mathbf{b}'_{12} + \alpha_3 \mathbf{b}'_{13} + \alpha_4 (\mathbf{b}'_{12} \times \mathbf{b}'_{13}) = \mathbf{g}' - \mathbf{b}'_1 \quad (16)$$

From (15), one infers

$$\mathbf{g} = \sum_{k=1}^3 \alpha_k \mathbf{b}_k + \alpha_4 (\mathbf{b}_{12} \times \mathbf{b}_{13}) \quad (17)$$

where $\alpha_1 = 1 - \alpha_2 - \alpha_3$. If G lies on the same plane identified by points B_1, B_2 and B_3 , α_4 could be set equal to zero and a simpler expression of \mathbf{g} would be achieved.

The relations in (13), (14) and (5) form a square system of 12 equations in 12 unknowns, namely $\mathbf{Y} = [\mathbf{X}^T, \boldsymbol{\tau}^T]^T = [\mathbf{x}_1^T, \mathbf{x}_2^T, \mathbf{x}_3^T, \boldsymbol{\tau}^T]^T$. All equations are algebraic and of degree at most 2 in the unknowns.

The number of unknowns may be reduced by eliminating cable tensions. Different strategies can be used, and these were all tested and compared in (Berti et al. 2013). However, the best parameterization proves to be the one described above, including cable tensions. This is mainly due to the reduced overestimation provided by the simpler expression of the equations, to the possibility of excluding negative-tension solutions from the search domain, and to more effective filtering procedures.

3.5. DGP for robots with four or five cables in tension

The DGPs for CDPRs with 4 and 5 cables in tension were analytically studied in (Abbasnejad and Carricato 2015), and they were proven to admit 216 and 140 solutions in the complex field, respectively.

Let the taut cables be labeled with $1, \dots, m$ ($m = 4$ or 5). In the following problem-solving procedure, three different parameterizations are adopted, depending on whether:

1. the platform anchor points $B_i, i = 1, \dots, m$, and G lie on the same plane;
2. the platform anchor points $B_i, i = 1, \dots, m$, lie on the same plane, but G does not;
3. The platform anchor points $B_i, i = 1, \dots, m$, do not lie on the same plane.

The variables \mathbf{X} used to describe the platform pose in case 1 are the position vectors of 3 non-aligned points of the platform, say $\mathbf{b}_1, \mathbf{b}_2$ and \mathbf{b}_3 . These points are called *reference points*. The remaining $m - 3$ anchor points and G are denoted as *secondary points*, and they are:

$$\begin{aligned} \mathbf{b}_i &= \sum_{k=1}^3 \beta_{ik} \mathbf{b}_k \quad i = 4, \dots, m \\ \mathbf{g} &= \sum_{k=1}^3 \alpha_k \mathbf{b}_k. \end{aligned} \quad (18)$$

where $\beta_{i1} = 1 - \beta_{i2} - \beta_{i3}$ and $\alpha_1 = 1 - \alpha_2 - \alpha_3$. The same approach is also adopted in case 2, except for the position vector of G , which is expressed using the relation in (17).

The parameters \mathbf{X} in case 3 are the position vectors of 4 points of the platform (not lying on the same plane). Similarly to case 1, the remaining secondary points can be expressed as:

$$\begin{aligned}\mathbf{b}_5 &= \sum_{k=1}^4 \beta_{ik} \mathbf{b}_k \\ \mathbf{g} &= \sum_{k=1}^4 \alpha_k \mathbf{b}_k.\end{aligned}\tag{19}$$

where $\beta_{i1} = 1 - \beta_{i2} - \beta_{i3} - \beta_{i4}$ and $\alpha_1 = 1 - \alpha_2 - \alpha_3 - \alpha_4$. The latter parameterization is over-redundant, since only three reference points would be sufficient to define the platform pose. However, using only 3 points leads to a more complex expression of the secondary points. The multiple occurrences of variables introduced by the cross product in (17) cause an overestimation in the evaluation of the secondary points, and consequently of the geometric-constraint equations. Indeed, the main reason for having three distinct parameterizations, specialized according to the geometry of the moving platform, is to be able to express each geometric constraint so as not to contain multiple occurrences of the unknown variables. Provided that l denotes the number of reference points, with $l \in \{3, 4\}$, the expression of the distance relations imposed by cable lengths are (Merlet 2004b):

$$(x_i - a_{ix})^2 + (y_i - a_{iy})^2 + (z_i - a_{iz})^2 = \rho_i^2 \quad i = 1, \dots, l \tag{20}$$

$$\left(\sum_{k=1}^l \beta_{ik} x_k - a_{ix} \right)^2 + \left(\sum_{k=1}^l \beta_{ik} y_k - a_{iy} \right)^2 + \left(\sum_{k=1}^l \beta_{ik} z_k - a_{iz} \right)^2 = \rho_i^2 \quad i = l+1, \dots, m \tag{21}$$

Similarly to (13), another set of C_2^l relations (namely, 3 for $l = 3$ and 6 for $l = 4$) emerges from the distance constraints between the reference points, i.e.

$$(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 = d_{ij}^2 \tag{22}$$

for $i, j \in \{1, \dots, l\}$ and $i \neq j$.

For CDPRs with 4 or 5 active cables, equations (20), (21) and (22) are not sufficient to determine the platform pose, but they must be solved together with relations (5) emerging from the mechanical equilibrium, thus adding cable tensions to the variable array, i.e. $\mathbf{Y} = [\mathbf{X}^T, \boldsymbol{\tau}^T]^T$, leading to a system of $m + C_2^l + 6$ equations in $3l + m$ unknowns (for $l = 3$ or 4, $C_2^l + 6 = 3l = 9$ or 12).

3.6. DGP for robots with six cables in tension

The DGP for this family of robots can be solved by means of the techniques developed for the Gough-Stewart Platform. Indeed, the 6 constraints in (4) imposed by the cable lengths in this case are sufficient to completely determine the platform pose. In the literature, there are plenty of studies about the direct kinematics of the Gough-Stewart Platform. In our code, the DGP of a CDPR with 6 active cables is solved by the interval method presented in (Merlet 2004b), and solutions with positive cable tensions are obtained by a filtering procedure (see Section 4).

4. Code structure

The most important parts of the interval-analysis code are discussed in this section. The code solves any geometry of robot suspended by $n \leq 6$ cables. In the current version, geometric parameters are assumed to be known, thus not affected by uncertainties. The ALIAS C++ library is used (Merlet 2007). It includes a large number of interval-analysis procedures

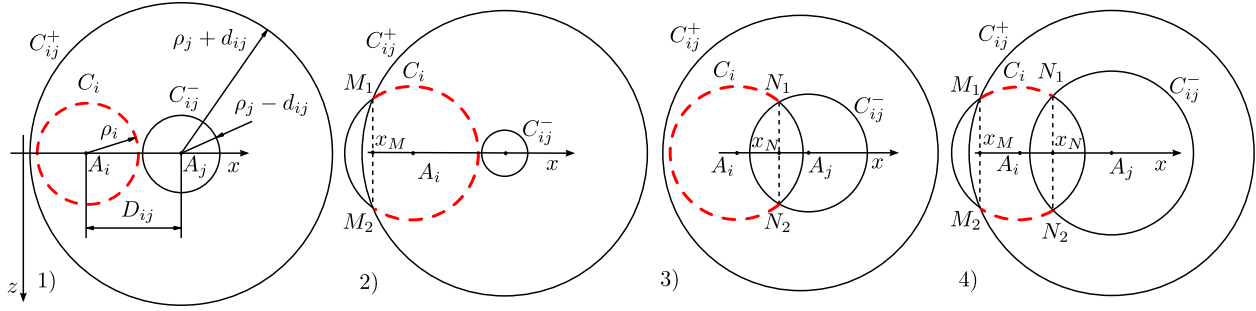


Fig. 5. The four different cases for the possible location of B_i as a function of the location of A_j , the leg length ρ_j and the distance between B_i and B_j . The admissible region zone for B_i is represented by the dashed line.

designed for solving systems of equations and inequalities, as well as optimization problems. ALIAS C++ relies on the BIAS/Profil library for basic interval arithmetics operations in double precision.

The main routine of the code reads the geometry data of the n - n CDPR at hand from a user-defined file, thus generating a list containing N_{DGP} sub-problems, each solving the DGP for a subset of m cables in tension, with $m \leq n$.

4.1. Initial search domain

A prerogative of any interval analysis method is that it requires an initial domain where solutions are searched for. For kinematics problems, this feature is an advantage, since it allows the physical constraints of the manipulator to be taken into account, and the code performances to be drastically improved.

Initial bounds for the pose variables. A simple method to obtain the initial bounds of the geometrical unknowns is the following. Let d_{ij} be the known distance between two reference points B_i and B_j , $i, j \in \{1, \dots, l\}$, and let the following quantities be defined:

$$\begin{aligned} X_{ij} &= a_{ix} + \rho_j + d_{ij} \\ Y_{ij} &= a_{iy} + \rho_j + d_{ij} \\ Z_{ij} &= a_{iz} + \rho_j + d_{ij} \end{aligned} \quad (23)$$

where $d_{ij} = 0$ when $i = j$. For each j , the coordinates of B_i must be (component-wise) greater than $[-X_{ij}, -Y_{ij}, -Z_{ij}]^T$ and lower than $[X_{ij}, Y_{ij}, Z_{ij}]^T$. Thus, the interval box containing the initial search space for reference point B_i is

$$\mathbf{B}_i = \begin{bmatrix} \left[\max_{j \in \{1, \dots, l\}} (-X_{ij}), \min_{j \in \{1, \dots, l\}} (X_{ij}) \right] \\ \left[\max_{j \in \{1, \dots, l\}} (-Y_{ij}), \min_{j \in \{1, \dots, l\}} (Y_{ij}) \right] \\ \left[\max_{j \in \{1, \dots, l\}} (-Z_{ij}), \min_{j \in \{1, \dots, l\}} (Z_{ij}) \right] \end{bmatrix} \quad (24)$$

The bounds expressed in (24) may be further improved. Consider a pair of cables i, j , and denote the distance between the exit points A_i and A_j with D_{ij} . If the i -th cable is taut, B_i lies on the surface of a sphere S_i centered in A_i , having radius ρ_i . Furthermore, B_i has to lie both inside the sphere S_{ij}^+ centered in A_j with radius $\rho_j + d_{ij}$ and outside the sphere S_{ij}^- centered in A_j with radius $\rho_j - d_{ij}$. Let, for the sake of simplicity, A_i and A_j be aligned on the x axis, and let C_i , C_{ij}^+ and C_{ij}^- be the projections of S_i , S_{ij}^+ and S_{ij}^- in plane xz , with M_1 , M_2 and N_1 , N_2 being the intersection points between C_i , C_{ij}^+ and C_i , C_{ij}^- , respectively. Figure 5 represents the following 4 cases:

1. if $\rho_j + d_{ij} > D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \leq D_{ij} - \rho_i$, B_i lies on S_i ;

2. if $\rho_j + d_{ij} < D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \leq D_{ij} - \rho_i$, B_i lies on the part of S_i contained within S_{ij}^+ ;
3. if $\rho_j + d_{ij} > D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \geq D_{ij} - \rho_i$, B_i lies on the part of S_i not contained within S_{ij}^- ;
4. if $\rho_j + d_{ij} < D_{ij} + \rho_i$ and $\max\{0, \rho_j - d_{ij}\} \geq D_{ij} - \rho_i$, B_i lies on the part of S_i contained within S_{ij}^+ but not contained within S_{ij}^- .

In the first case, no improvement on the bounds of B_i can be obtained with respect to those determined in (24). In the second case, x_M , which is the x coordinate of points M_1 and M_2 , is a better lower bound for the first component of \mathbf{B}_i , compared to $-X_{ij}$. In addition, if $x_M - a_{ix} \geq 0$, also the third component of \mathbf{B}_i can be updated to $[-z_M, z_M]$. Similarly, in the third case, x_N is a better upper bound for the first component of \mathbf{B}_i than X_{ij} , and if $x_N - a_{ix} \leq 0$ the third component of \mathbf{B}_i can be narrowed to $[-z_N, z_N]$. In the fourth case, both x_M and x_N improve the initial bounds for \mathbf{B}_i . Furthermore, thanks to the spherical symmetry of the problem, $[-z_M, z_M]$ and $[-z_N, z_N]$ can be used to improve the bounds for the component y_i of \mathbf{B}_i as well. If A_i and A_j are not aligned with the x axis, a rotation matrix always exists that allows the bounds determined by this procedure to be conveniently rotated, thus improving the actual box \mathbf{B}_i . For the DGP of a n - n CDPR with m cables in tension, the above procedure has to be repeated for all pairs of taut cables i, j with $i, j \in \{1, \dots, m\}$. It is possible to further improve the initial bounds for B_i , $i \in \{1, \dots, m\}$, by applying this procedure also for each slack cable k , with $k \in \{m+1, \dots, n-m\}$ provided that the radius of S_{ik}^- is set equal to zero. Though extremely simple, this latter refinement may largely speed up the computation, since it sometimes determines an empty initial search space, thus preventing any further calculation.

Initial bounds for cable tensions. The tension variables in $\boldsymbol{\tau}$ require an initial search domain, except when $n = 6$. On the one hand, this can be an advantage since the lower bound for each element of $\boldsymbol{\tau}$ can be set equal to zero, so that only solutions with nonnegative cable tensions are looked for. On the other hand, choosing a priori an upper bound for cable tensions is not an easy task. For the particular case of suspended cable robots, an upper bound for tensions can be obtained by observing that each element of the third line of matrix \mathbf{M} in (5) is negative, i.e. $M_{3j} < 0$, for $j = 1, \dots, m$. Thus, by considering the third equation of (5), namely

$$Q = - \sum_{j=1}^m M_{3j} \frac{\tau_j}{\rho_j} \quad (25)$$

the following inequality holds:

$$\tau_j \leq Q \rho_j / (-M_{3j}) \quad j = 1, \dots, m \quad (26)$$

where $(-M_{3j})$ is the lower bound of $-M_{3j}$.

However, for all CDPRs which are not suspended, an upper bound for tensions cannot be easily determined. Setting a too small value can cause the algorithm to exclude feasible solutions (with high values of cable tensions), that may actually exist within the geometrical workspace. These solutions are the most dangerous from the mechanical point of view and it is important to determine them, since they may possibly occur. Setting a too high value (compared with the intensity Q of the applied load) infers a large interval evaluation of static equations (5), and thus a large number of bisections in order to assess if mechanical equilibrium is attained. This leads to large computation times.

In Section 4.3, a specific filtering procedure is described, which can drastically reduce the influence of the width of the cable-tension search domain on the code performances. Thanks to this improvement, an initial search domain for cable tensions is no longer required and, when not provided, the algorithm automatically finds all feasible solutions.

Cable numbering. For a generic n - n CDPR, with $n > l$, the l reference points on the platform may be chosen arbitrarily. However, this choice has an influence on both the size of the geometric search space and the value of coefficients α_k and β_{ik} of the secondary points (cf. equations (17) through (19)), consequently increasing or decreasing their interval evaluation.

As shown in (Merlet 2004b) for the Gough-Stewart platform, these two factors, have a great influence on the computation speed.

In order to asses the performance of a given combination of reference points, the following index is defined (for each sub-DGP) on an empirical basis:

$$\Phi = 5I_\alpha + 3I_{min} + 2I_{Vl} + I_{Vm} \quad (27)$$

where I_α , I_{min} , I_{Vl} and I_{Vm} are coefficients, comprised between 0 and 1. I_α is the (normalized) mean value of the absolute value of all coefficients α_k and β_{ik} , I_{min} is the (normalized) diameter of the interval having the minimal width in the geometrical search domain, I_{Vl} and I_{Vm} are, respectively, the (normalized) mean widths of the components forming the initial search domain for the l reference points and for the m active cables. On the basis of (27), the algorithm ranks all combinations of reference points and picks the one having the lowest Φ . The weights 5, 3, 2 and 1 assigned to, respectively, I_α , I_{min} , I_{Vl} and I_{Vm} were chosen, after several tests, in order to provide the combination of reference points generally leading to the best computation time.

4.2. Evaluation operator

The evaluation operator \mathcal{E} is implemented by means of the ALIAS procedure `Solve_General_Gradient_Interval` (SGGI). Since the functions involved in Eqs. (5,10,11,13,14,20,21,22) are at least of class C^2 , the Jacobian matrix of the system can be analytically computed. Accordingly, the SGGI improves the interval evaluation of the aforementioned functions by conveniently using gradients, and by taking advantage of possible monotonicities (Merlet 2007). Moreover, SGGI exploits derivatives by applying, during the evaluation process, a global filtering method based on the classic interval Newton scheme, as described in (Ratschek and Rokne 1995).

Unicity of the solution. Since second-order derivatives can be analytically computed, the operator \mathcal{E} applies a procedure based on Kantorovitch theorem (Tapia 1971) to each box that is deemed to be a solution, in order to check whether it contains a unique solution or not. Let any sub-DGP of h equations $\mathbf{f} = \mathbf{0}$ in h unknowns \mathbf{Y} among those presented in Section 3 be considered. Let \mathbf{Y}_0 be a robot configuration, r_0 , s_0 and p positive real constants, $\mathbf{U} = \{\mathbf{Y} : \|\mathbf{Y} - \mathbf{Y}_0\|_\infty \leq 2r_0\}$ a ball centered in \mathbf{Y}_0 , and $\|\mathbf{A}\|_\infty = \max_i \sum_j |a_{ij}|$ be the maximum absolute row sum norm of a generic matrix \mathbf{A} . If \mathbf{Y}_0 is such that

1. the Jacobian of the system of equations has an inverse $\mathbf{\Gamma}_0$ in \mathbf{Y}_0 such that $\|\mathbf{\Gamma}_0\|_\infty \leq s_0$,
2. $\|\mathbf{\Gamma}_0 \mathbf{f}(\mathbf{Y}_0)\|_\infty \leq r_0$,
3. $\sum_{k=1}^h \left| \frac{\partial^2 f_i(\mathbf{Y})}{\partial Y_j \partial Y_k} \right| \leq p$ for $i, j = 1, \dots, h$ and $\mathbf{Y} \in \mathbf{U}$,
4. the constants s_0 , r_0 , p satisfy $2hs_0r_0p \leq 1$,

then the system of equation $\mathbf{f} = \mathbf{0}$ admits a unique solution \mathbf{Y}^* in \mathbf{U} , and the Newton method used with \mathbf{Y}_0 as the initial estimation of the solution will converge toward \mathbf{Y}^* . By this algorithm, the code can guarantee that the solution is unique in the given box, and it can calculate it with an arbitrary accuracy. The implementation of the Kantorovitch procedure relies on the fact that the equations involved in all DGPs are at most of degree 2 in the unknowns, so that second-order derivatives are constants and p can be precomputed in the initialization of the algorithm (Merlet 2004b). Thus, the computational burden of this procedure is very small. The Kantorovitch procedure fails only when, with the current computer precision, the value of at least one equation cannot be determined over a box reduced to a point; in this case, it may be shown that no other algorithm is capable of solving the system.

Stability. Each solution identified by \mathcal{E} is eventually processed by a procedure that determines whether the corresponding equilibrium configuration is stable or not. A sufficient condition for the equilibrium to be stable consists in having the (reduced) Hessian matrix \mathbf{H}_r being positive definite. If \mathbf{I}_3 denotes the 3×3 identity matrix, and $\tilde{\mathbf{n}}$ denotes, for a generic vector \mathbf{n} , the skew-symmetric matrix associated with the operator $\mathbf{n} \times$, \mathbf{H}_r may be computed as (Carricato and Merlet 2013)

$$\mathbf{H}_r = \mathbf{N}^T \mathbf{H} \mathbf{N}, \quad (28)$$

where

$$\mathbf{H} = \sum_{i=1}^m \frac{\tau_i}{\rho_i} \begin{bmatrix} \mathbf{I}_3 & -(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}}) \\ (\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}}) & \frac{1}{2} [(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}})(\tilde{\mathbf{g}} - \tilde{\mathbf{a}}_i) + (\tilde{\mathbf{g}} - \tilde{\mathbf{a}}_i)(\tilde{\mathbf{b}}_i - \tilde{\mathbf{g}})] \end{bmatrix}, \quad (29)$$

and \mathbf{N} is a $6 \times (6 - m)$ matrix whose columns generate the null space of the Jacobian matrix \mathbf{J}^T :

$$\mathbf{J}^T = \begin{bmatrix} (\mathbf{b}_1 - \mathbf{a}_1)^T / \rho_1 & (\mathbf{b}_1 - \mathbf{g}) \times (\mathbf{b}_1 - \mathbf{a}_1)^T / \rho_1 \\ \vdots & \vdots \\ (\mathbf{b}_m - \mathbf{a}_m)^T / \rho_m & (\mathbf{b}_m - \mathbf{g}) \times (\mathbf{b}_m - \mathbf{a}_m)^T / \rho_m \end{bmatrix}. \quad (30)$$

in this case computed assuming G as the moment pole.

4.3. Filter operator

The filter operator \mathcal{F} comprises different procedures. Each of them takes an interval box as an input and tries either to improve its bounds or to eliminate it by using *constraint propagation* techniques (Jaulin 2001). Most filtering functions included in \mathcal{F} rely on the 2B approach. This method consists in rewriting an equation as the equality of two terms, determining if the interval evaluations of both terms are consistent and, if not, using consistency to improve the width of the interval for one or more unknowns. For instance, let the equation $x^2 - 2x + 1 = 0$ be considered. By introducing the new variable $X = x^2$, the original equation may be re-written as $X = 2x - 1$. Now, let $[\underline{u}, \bar{u}]$ be the interval evaluation of $2x - 1$. If $\bar{u} > 0$, then the inverse function of X indicates that x should lie in $[-\sqrt{\bar{u}}, \sqrt{\bar{u}}]$ and, from this information, the current interval of x may be updated. If $\underline{u} > 0$, the inverse function of X shows that x should lie outside $[-\sqrt{\underline{u}}, \sqrt{\underline{u}}]$: if the range of x is included in this interval, then there is no solution of the equation in the current box. The 2B approach can also be applied, with some modifications, to constraints expressed by inequalities.

Filters on geometrical constraints. Thanks to the adopted parameterization, the 2B method applied to the geometrical constraints expressed in (20) is very effective. Indeed, the 2B method reduces the box \mathbf{B}_i containing the intervals of coordinates x_i , y_i and z_i to the smallest box containing the intersection between \mathbf{B}_i itself and the surface of the sphere centered in A_i with radius ρ_i . The same scheme is applied for the constraints expressed by the point-to-point distance relations (22).

When dealing with equations involving secondary points, such as (21), first the interval components X_j , Y_j and Z_j of the j -th secondary point are computed by interval evaluating relations (18) and (19); then, the 2B method is applied to all geometrical relations where secondary points are involved, namely (21) and the distance relations between the secondary points and the reference ones. If any improvement is achieved for X_i , Y_i and Z_i , the 2B method is applied back to (18) and (19).

Since the 2B approach can be applied to inequalities too, the following constraints are also considered and processed:

$$\begin{aligned} |\mathbf{b}_i - \mathbf{a}_j|^2 &\leq (\rho_j + d_{ij})^2 \\ |\mathbf{b}_i - \mathbf{a}_j|^2 &\geq (\rho_j - d_{ij})^2 \end{aligned} \quad (31)$$

for any $i \neq j$.

When a sub-DGP is analyzed, further inequalities are added to the filtering procedure, in order to take the constraints imposed by slack cables into account, namely:

$$\begin{aligned} |\mathbf{b}_i - \mathbf{a}_i|^2 &\leq \rho_i^2 \\ |\mathbf{b}_j - \mathbf{a}_i|^2 &\leq (\rho_i + d_{ij})^2 \end{aligned} \quad (32)$$

where i denotes one of the $n - m$ cables considered to be slack and j denotes one of the m active cables.

Another procedure that helps to improve the bounds on the pose variables consists in applying the 2B method to all relations like (18) and (19) that emerge from all possible choices of reference and secondary points (not necessarily the choice that minimizes function Φ in (27)). Finally, the 2B approach can be applied to the geometric constraint:

$$(\mathbf{b}_j - \mathbf{b}_i) \cdot (\mathbf{b}_k - \mathbf{b}_i) = \varphi_{ijk} \quad (33)$$

where $i, j, k \in 1, \dots, n$, $i \neq j \neq k \neq i$, and $\varphi_{ijk} = \cos(\widehat{B_j B_i B_k})$. The latter is a known constant deriving from the geometry of the moving platform.

Filters on static constraints. For CDPRs having $m \leq 5$ active cables, the 2B method may be applied to the equilibrium equations expressed by (5). Moreover, rewriting matrix \mathbf{M} by choosing another exit point on the base as the moment pole, three new equations may be obtained from the last three rows of \mathbf{M} . Repeating this process for all base exit points, $3(m - 1)$ additional relations can be used in the filter, thus being processed with the 2B approach. Even if this filtering technique is quite effective, it does not reduce the negative influence that large initial bounds on cable tension may have on the code performances. So, another routine is implemented that nearly eliminates this dependence.

The main idea consists in trying to solve the interval linear system obtained by rewriting (5) as:

$$\underbrace{\begin{bmatrix} \mathcal{L}_1/\rho_1 & \dots & \mathcal{L}_m/\rho_m \end{bmatrix}}_{\mathbf{J}} \boldsymbol{\tau} = \underbrace{-\mathbf{Q}\mathcal{L}_e}_{\mathbf{Q}} \quad (34)$$

where \mathbf{J} is a $6 \times m$ interval matrix and \mathbf{Q} an interval vector. When $m \leq 5$, system (34) is not square and admits a solution only when $\text{rank}(\mathbf{J}) \leq m$. An approximate solution of this overdetermined system of linear interval equations may be obtained by left-multiplying both members of (34) by an arbitrary $m \times 6$ matrix \mathbf{K} (Rohn 1996), namely

$$\underbrace{\mathbf{K}\mathbf{J}}_{\mathbf{C}} \boldsymbol{\tau} = \mathbf{K}\mathbf{Q} \quad (35)$$

By doing so, a square system is obtained and the interval version of the Gauss elimination scheme provided by ALIAS can be applied (Merlet 2007). This algorithm, when successful (i.e. when \mathbf{C} contains only regular matrices), returns an interval vector that is the enclosure of the infinite possible solutions of the linear interval system. According to (Rohn 1996), in order to improve the algorithm performances, a good candidate for \mathbf{K} is the Moore-Penrose pseudoinverse of the mid-point evaluation of \mathbf{J} , i.e.

$$\mathbf{K} = (\text{mid}(\mathbf{J})^T \text{mid}(\mathbf{J}))^{-1} \text{mid}(\mathbf{J})^T \quad (36)$$

However, this choice is questionable in this case, mainly due to the product $\text{mid}(\mathbf{J})^T \text{mid}(\mathbf{J})$. Indeed, while the first three elements of each column of \mathbf{J} are pure numbers, the remaining three are the result of the cross product between a dimensionless vector and a position vector whose components are lengths. Accordingly, the elements of the matrix $\text{mid}(\mathbf{J})^T \text{mid}(\mathbf{J})$ contains the sum between pure numbers and lengths, which is physically meaningless.

We solve this issue by considering a generalized inverse defined as

$$\mathbf{K} = (\mathbf{W}^T \mathbf{J})^{-1} \mathbf{W}^T \quad (37)$$

where \mathbf{W} is a $6 \times m$ matrix having the first (last) 3 rows equal to the last (first) 3 rows of \mathbf{J} . \mathbf{K} is a true generalized inverse matrix, since it satisfies the Penrose conditions $\mathbf{J}\mathbf{K}\mathbf{J} = \mathbf{J}$ and $\mathbf{K}\mathbf{J}\mathbf{K} = \mathbf{K}$ (Penrose 1955). By adopting \mathbf{W} , all operations contained in (37) are consistent in terms of units. Moreover, this choice still implies that, as the box size decreases, \mathbf{C} tends to be the identity matrix, which is the best input for the Gauss elimination scheme. It must be remarked that in order to reduce the overestimation introduced by the matrix product, each element of \mathbf{C} can be rewritten trying to obtain the lowest number of multiple occurrences of the pose parameters.

A further refinement comes from considering that each element of the interval matrix \mathbf{C} has the following form

$$C_{ij} = \sum_{k=1}^6 K_{ik} J_{kj} \quad (38)$$

and that each column of \mathbf{J} is a Plücker vector, so that

$$J_{1j}^2 + J_{2j}^2 + J_{3j}^2 - 1 = 0 \quad (39)$$

$$J_{1j} J_{4j} + J_{2j} J_{5j} + J_{3j} J_{6j} = 0 \quad (40)$$

for $j = 1, \dots, 6$. Thus, the bounds for each element of \mathbf{C} may be improved by searching the minimum and the maximum of the function

$$C'_{ij}(\mathbf{q}) = \sum_{k=1}^6 K_{ik} q_k \quad (41)$$

where \mathbf{q} must satisfy the following constraints:

$$q_1^2 + q_2^2 + q_3^2 - 1 = 0$$

$$q_1 q_4 + q_2 q_5 + q_3 q_6 = 0 \quad (42)$$

$$\underline{M}_{kj} \leq q_k \leq \overline{M}_{kj} \quad k = 1, \dots, 6$$

The optimization problem emerging from relations (41) and (42) can be analytically solved with the generalized Lagrange multiplier method and, thus, implemented without using numeric optimization algorithms.

By virtue of the optimization of \mathbf{C} , improvements on the bounds of cable tension variables can be obtained even if cable tensions are not involved in the optimization formulation. When CDPRs having $m \leq 5$ are considered, this strategy allows the filtering procedure to partially reduce the coupling between geometric and tension variables and, as a consequence, the influence of the width of the tension search domain on computation time. When dealing with the DGP of a CDPR with 6 taut cables, cable tensions are not included in the parameterization, and the above procedure is applied to discard boxes having negative cable tensions. In this case $\mathbf{K} = \text{mid}(\mathbf{J})^{-1}$, since the linear system (34) is square.

Influence on the bisection process The effectiveness of the filtering procedures on the static constraints, especially the one relying on the Gauss elimination scheme, allows one to slightly modify the bisection process. Indeed, rather than bisecting the largest variable contained in $\mathbf{Y} = [\mathbf{X}^T, \boldsymbol{\tau}^T]^T$, it is more convenient to bisect the largest variable contained in \mathbf{X} until $\max(w(\mathbf{X})) \geq \varepsilon$ (where ε is the threshold used by operator \mathcal{E} to stop the bisection process, see Sec. 2), and to let \mathcal{F} filter the variables contained in $\boldsymbol{\tau}$. For boxes having $\max(w(\mathbf{X})) < \varepsilon$, the bisection process picks the variable having the largest width all over \mathbf{Y} .

The advantages provided by this change on the bisection strategy are:

- a lower number of bisections
- a minimal influence of the width of the initial search domain of cable tension variables on code performances
- small differences between the computation times for the DGP of robots with 3, 4 or 5 cables in tension, even though they involve a different number of variables.

4.4. Parallel implementation

Most interval-analysis-based algorithms are appropriate for a distributed implementation, since processing a given box does not generally depend on processing of the other boxes in the list. A master computer manages the list and it sends a box to a slave computer. The slave executes the algorithm, by performing a few bisections. Then, it returns the remaining boxes to the master, and it waits for a new box to process. Such a scheme may be easily implemented in a network of workstations. The decrease of computation time will be, in general, less than proportional to the number of slaves, due to the overhead of data transmission between the master and the slaves.

This approach may also take advantage of modern multi-core CPU architectures. By following this scheme and by using POSIX thread libraries, a distributed implementation of the DGP code was set up on a single workstation with a multi-core CPU. In the first step, an instance of \mathcal{E} generates a few boxes and it stores them in the list \mathcal{L} . Then, a number of threads equal to the number of CPUs is created, with each one taking a box from \mathcal{L} . A local instance of \mathcal{E} performs an assigned number of bisections and it appends the generated boxes to \mathcal{L} . The solutions that are found, if any, are appended to the solution list \mathcal{S} .

5. Testing

In this section, some tests are reported to show the performances of the problem-solving code. In all examples, lengths are expressed in meters and angles in radians. Each solution is given as

- a configuration number;
- the number m of taut cables;
- 3 parameters $x_{O'}$, $y_{O'}$ and $z_{O'}$ describing the position of the origin O' of the mobile frame in $Oxyz$;
- 3 angles φ_x , φ_y and φ_z , representing the orientation of the platform by three consecutive rotations along the x , y and z fixed axes;
- n values of cable tensions;
- a symbol that denotes if the solution is stable ($>$) or not ($<$).

All tests were performed on a PC having an Intel® Xeon W3520 2.67 GHz CPU with 4 cores.

5.1. Test 1

The first test (Table 1) is taken from Berti et al. (2013). The aim is to compare the performances of the current code with respect to the original implementation, thus showing the improvements obtained in this version of the problem-solving routine.

The algorithm presented in (Berti et al. 2013), with the same parameterization described in Section 3, finds all equilibrium configurations with 3 cables in tension in 202 seconds, but it does not verify whether solutions with slack cables exist and

does not assess stability. The current version of the code analyzes all possible equilibrium configurations with m cables in tension, $m \leq 3$, in roughly 25 seconds, including stability analysis. Both computation times refer to single-core executions.

Table 1. Solution set for a CDPR suspended by 3 cables.

DATA											
$\mathbf{a}_1 = [0, 0, 0]$ $\mathbf{a}_2 = [10, 0, 0]$ $\mathbf{a}_3 = [0, 12, 0]$ $\mathbf{b}'_1 = [0.817, 0, 0]$ $\mathbf{b}'_2 = [-0.408, 0.707, 0]$ $\mathbf{b}'_3 = [-0.408, -0.707, 0]$ $\mathbf{g}' = [0, 0, -0.577]$ $(\rho_1, \rho_2, \rho_3) = [7.5, 10, 9.5]$ $Q = 1$											
Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	Stability
1	3	2.745	3.979	5.506	3.007	0.340	0.109	0.526	0.511	0.581	>
2	3	1.700	3.687	5.809	0.339	-1.036	-2.596	0.676	0.251	0.486	<
3	3	3.020	4.757	3.879	-0.038	0.027	0.776	0.590	0.783	0.956	<
4	3	1.846	4.074	5.322	2.146	-0.708	2.423	0.684	0.305	0.614	<
5	3	2.138	4.287	6.030	-0.482	-0.360	-2.211	0.546	0.325	0.550	<
6	3	3.499	5.369	4.709	-2.908	-0.174	-2.659	0.289	0.787	0.912	<
—	2	-	-	-	-	-	-	-	-	-	-
—	1	-	-	-	-	-	-	-	-	-	-

5.2. Test 2

This second test (Table 2) shows the case of a 6-6 CDPR having all cable anchor points on the moving platform lying on the same plane. As shown in Section 3, a specific parameterization can be used in this case. The algorithm finds 2 solutions with 6 cables in tension, 11 solutions with 5 taut cables, 12 with 4 and 9 with 3 in 187 seconds by a single-core execution, and in 49 seconds if the code is distributed over 4 CPUs. Five out of 34 solutions are stable. Only stable solutions are reported in Table 2, for the sake of conciseness. Their feasibility was checked on a small prototype employed for teaching purposes (Fig. 6).

Table 2. Solution set for a CDPR suspended by 6 cables: all cable anchor points on the platform are coplanar.

DATA														
$\mathbf{a}_1 = [0.238, 0.063, 0.785]$ $\mathbf{a}_2 = [0.736, 0.551, 0.785]$ $\mathbf{a}_3 = [0.738, 0.551, 0.781]$ $\mathbf{a}_4 = [0.062, 0.729, 0.785]$ $\mathbf{a}_5 = [0.064, 0.729, 0.789]$ $\mathbf{a}_6 = [0.238, 0.062, 0.785]$ $\mathbf{b}'_1 = [0.369, -0.002, 0.025]$ $\mathbf{b}'_2 = [0.369, 0.002, 0.025]$ $\mathbf{b}'_3 = [-0.017, 0.033, 0.025]$ $\mathbf{b}'_4 = [-0.020, 0.031, 0.025]$ $\mathbf{b}'_5 = [-0.020, -0.031, 0.025]$ $\mathbf{b}'_6 = [-0.017, -0.033, 0.025]$ $\mathbf{g}' = [0, 0, 0.019]$ $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6) = [0.799, 0.790, 0.796, 0.787, 0.798, 0.791]$ $Q = 1$														
Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	Stability
1	6	0.346	0.453	0.066	-0.003	0.005	-0.518	0.153	0.228	0.156	0.225	0.163	0.219	>
2	6	0.348	0.453	0.088	-0.019	0.004	2.081	0.231	0.165	0.233	0.162	0.239	0.155	>
3	5	0.363	0.454	0.122	-2.420	0.406	-1.341	0.202	0.333	0.105	0.362	0.000	0.151	>
4	5	0.341	0.437	0.122	2.419	0.345	0.449	0.100	0.362	0.000	0.159	0.204	0.327	>
5	5	0.335	0.464	0.122	-3.124	-0.823	2.636	0.000	0.149	0.210	0.326	0.116	0.352	>

5.3. Test 3

Test 3 (Table 3) refers to the large-scale robot MARIONET-VR (Merlet 2010). This robot uses linear actuators instead of winches to change the cable lengths, and it is employed in a virtual-reality environment as a motion provider and haptic

device. The moving platform was designed for demonstration purposes and its anchor points do not belong to the same plane. Table 3 reports all stable solutions found by running the code for all possible sets of taut cables, i.e. for 63 sub-problems. The number of solutions retrieved is 17, i.e. 3 with 6 cables in tension, 1 with 5 cables in tension, 7 with 4 cables in tension, 4 with 3 cables in tension and 2 with 2 cables in tension. The computation time is 702 seconds by a single-core execution, and 178 seconds if the code is distributed over 4 CPUs. Feasibility was checked on the prototype shown in Fig. 7.

Table 3. Solution set for a CDPR suspended by 6 cables: cable anchor points on the platform are not coplanar.

DATA														
$\mathbf{a}_1 = [1.886, 0.558, 2.590]$ $\mathbf{a}_2 = [1.921, -2.876, 2.750]$ $\mathbf{a}_3 = [-1.498, 0.809, 2.745]$ $\mathbf{a}_4 = [1.567, 0.895, 2.600]$ $\mathbf{a}_5 = [-1.581, -3.015, 2.740]$ $\mathbf{a}_6 = [-1.692, 0.372, 2.580]$ $\mathbf{b}'_1 = [0.63, 0.59, 0.585]$ $\mathbf{b}'_2 = [0.63, 0.585, 0.015]$ $\mathbf{b}'_3 = [-0.03, 0.05, 0.585]$ $\mathbf{b}'_4 = [0.63, 0.05, 0.585]$ $\mathbf{b}'_5 = [-0.03, 0.585, 0.015]$ $\mathbf{b}'_6 = [-0.03, 0.59, 0.585]$ $\mathbf{g}' = [0.3, 0.3, 0.3]$ $(\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6) = [2.755, 3.519, 2.849, 2.837, 3.489, 2.609]$ $Q = 1$														
Conf.	m	$x_{O'}$	$y_{O'}$	$z_{O'}$	φ_x	φ_y	φ_z	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	Stability
1	6	-0.270	0.235	0.778	2.554	0.124	0.080	0.398	0.226	0.248	0.078	0.244	0.268	>
2	6	0.253	-0.520	0.338	0.960	-0.105	-3.077	0.262	0.291	0.293	0.278	0.314	0.283	>
3	6	-0.278	-1.470	0.549	-0.670	0.014	-0.043	0.374	0.271	0.156	0.004	0.376	0.220	>
4	5	-0.279	-1.470	0.549	-0.669	0.016	-0.046	0.381	0.267	0.161	0.000	0.380	0.213	>

The tests reported above are representative of the average performance of the problem-solving code. The first test shows that the refinements introduced in this version lead to computation times that are roughly four times smaller compared to the preliminary results presented in (Berti et al. 2013). The second and the third test present the complete analysis of the DGP of a 6-6 CDPR. For the former test, thanks to planar disposition of the anchor points on the platform, the code takes advantage of simpler parameterizations for the sub-problems with 4, 5 and 6 cables, as explained in Section 3, with a strong effect on computation times. The third test analyzes the DGP of a robot with a spatial disposition of the anchor points on the moving platform. For this kind of geometry, the code needs higher computation times. However, performances are still particularly interesting: without considering the 21 subproblems with 1 and 2 active cables (they are generally processed in less than 1 second), the code solves 20 sub-DGP with 3 cables in tension, 15 with 4 cables in tension, 6 with 5 cables in tension and 1 with 6 cables in tension in 702 seconds, with a mean computation time of roughly 17 seconds for each problem. It is worth mentioning that the obtained computation time is not necessarily the lowest one. It can happen that the procedure that selects the best cable numbering for each DGP fails in finding the best combination. Indeed, the ranking criterion expressed in (41) is an empirical relation coming from the data gathered during tests and on the results obtained in (Merlet 2004b), which deserve further investigation.

In both tests 2 and 3, multiple feasible solutions are obtained for a given set of cable lengths. In the second test, if the platform is in configuration No. 1 (Fig. 6a), a potential external disturbance can cause the robot to fall in one of the 3 solutions with 5 cables in tension (Fig. 6b shows configuration No. 3), and without additional external sensors it is difficult to determine in which solution branch the robot is operating. Conversely, configuration No. 2, with 6 cables in tension, is not directly reachable with a finite movement of the platform without detaching and reattaching cables.

In the third example, the robot (Fig. 7a) can easily switch from configuration No. 1 to configuration No. 3 (Fig. 7b) (passing through an unstable configuration with 4 taut cables) and vice-versa, while configuration No. 2 cannot be directly reached without detaching cables. Configuration No. 3 with 6 cables in tension is almost coincident with configuration No. 4, with 5 taut cables, as can be noted by the small value of tension in the fourth cable. Accordingly, in this part of the workspace, the robot may easily act as an underconstrained one in an uncontrolled way.

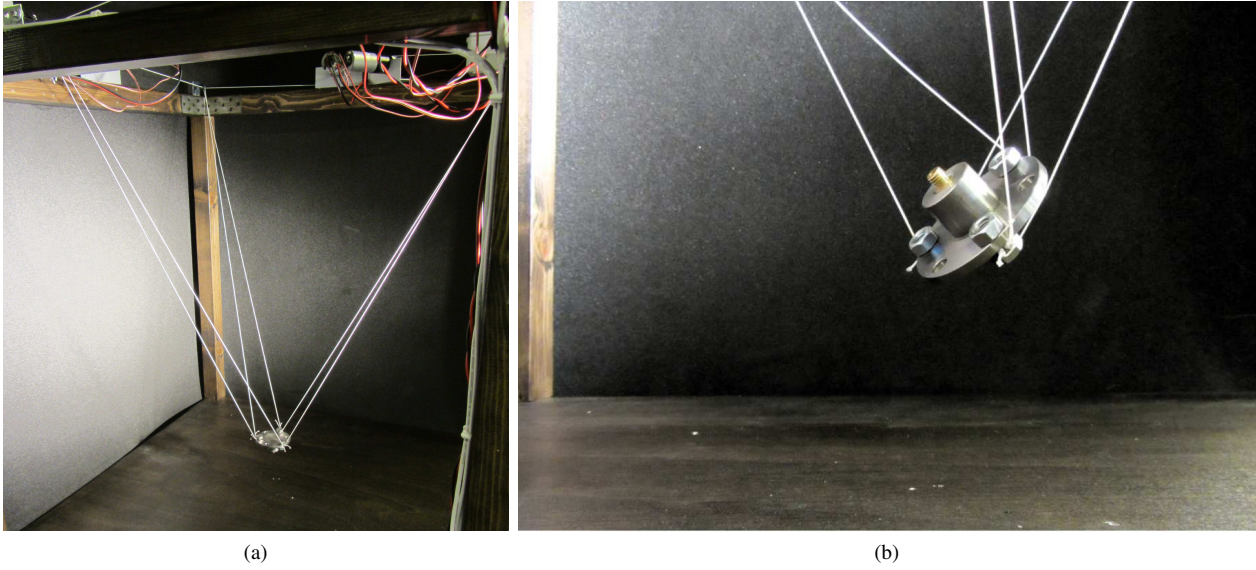


Fig. 6. Validation of Test 2 on a teaching prototype: (a) configuration No. 1 with 6 taut cables; (b) configuration No. 3 with 5 cables in tension.

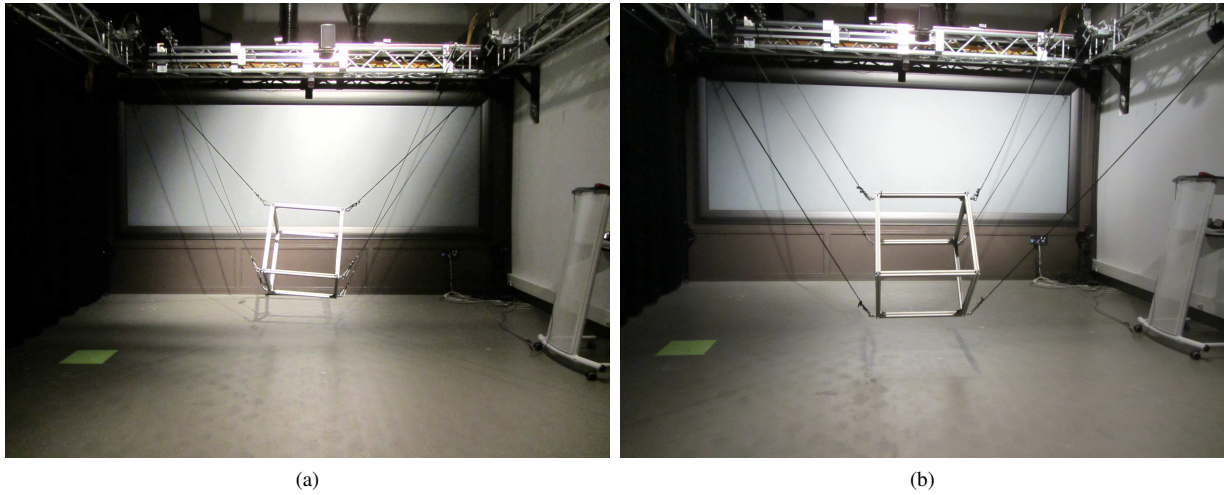


Fig. 7. Validation of Test 3 on the large scale robot MARIONET-VR: (a) configuration No. 1 with 6 cables in tension; (b) configuration No. 3 with 6 cables in tension.

The current version of the code can determine the stability of equilibrium only after the computation of all statically-admissible solutions. The results obtained during the experimental campaign show that stable solutions are usually a small subset of statically-admissible poses. Therefore, the implementation of a stability filter during the computation would allow the domain regions where unstable configuration may occur to be discarded, thus allowing the direct computation of stable solutions. A few attempts have been made in order to develop such a filter, but the overestimation introduced by interval-analysis computation severely compromises its effectiveness. On the other hand, the analysis of unstable configurations may be of practical interest. Indeed, an unstable solution may provide a bridge towards stable configurations. It is specific to CDPRs, indeed, the possibility of configuration changes without crossing singularities. In this situation, the set of physical constraints represented by the interference between the platform and taut cables may be reduced, thus disclosing new paths for other operating modes.

6. Conclusions

An interval-analysis-based code for the solution of the direct geometrico-static problem (DGP) of cable-driven parallel robots (CDPRs) was presented. The algorithm computes all solutions of the DGP for a generic CDPR, including those with slack cables, with very interesting computation times. Interval analysis guarantees that these solutions are certified against numerical errors, and allows the specific constraints given by these robots to be taken into account, like for example the positiveness of cable tensions. The tests conducted so far showed that, even with a simplified cable model, the code can give accurate results.

The code can not be used, as it is, as a real-time module in control routines. On the other hand, in real-time applications, when the pose of the robot at the time t_0 is known, it is possible to reduce the initial search-domain to the box enclosing all poses reachable by the robot in a single cycle-time t_i . With this strategy, the interval approach can be effectively applied to solve the DGP with a single configuration of taut cables in real-time (Merlet 2004b), but it cannot be used to compute the complete solution set, thus including different sets of taut cables. However, the current code still has room for improvements, for example exploiting the excellent characteristics that the procedure has in parallel implementation, and further improving the code efficiency.

Further enhancements may also be applied to improve accuracy of results. The first one consists in considering the errors affecting the measures of the position of the cable exit points on the base and of the anchor points on the platform, as well as the errors on cable lengths. Other improvements consist in considering the effects that cable mass and elasticity have on the platform pose and in checking solutions against possible interferences between cables or between cables and the robot platform or structure. All these issues can be dealt with by interval analysis and they are currently under investigation (Blanchet and Merlet 2014, Merlet and Alexandre-dit Sandretto 2015).

Acknowledgments

The financial support of the Italian Ministry of Education, Universities and Research through the PRIN grant No. 20124SMZ88_002 is gratefully acknowledged.

References

- Abbasnejad, G. and Carricato, M. (2012). Real solutions of the direct geometrico-static problem of under-constrained cable-driven parallel robots with 3 cables: a numerical investigation. *Meccanica*, 47(7):1761–1773.
- Abbasnejad, G. and Carricato, M. (2015). Direct geometrico-static problem of underconstrained cable-driven parallel robots with n cables. *IEEE Transactions on Robotics*, 31(2):468–478.
- Albus, J., Bostelman, R., and Dagalakakis, N. (1993). The Nist robocrane. *Journal of Robotic Systems*, 10(5):709–724.
- Behzadipour, S. and Khajepour, A. (2006). Stiffness of cable-based parallel manipulators with application to stability analysis. *Journal of Mechanical Design*, 128(1):303–310.
- Berti, A., Merlet, J.-P., and Carricato, M. (2013). Solving the direct geometrico-static problem of 3-3 cable-driven parallel robots by interval analysis: Preliminary results. In Bruckmann, T. and Pott, A., editors, *Cable-Driven Parallel Robots*, volume 12 of *Mechanisms and Machine Science*, pages 251–268. Springer Berlin Heidelberg.
- Blanchet, L. and Merlet, J.-P. (2014). Interference detection for cable-driven parallel robots (CDPRs). In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Besançon, France, July 8-11, 2014*, pages 1413–1418.
- Bosscher, P., Riechel, A. T., and Ebert-Uphoff, I. (2006). Wrench-feasible workspace generation for cable-driven robots. *IEEE Transactions on Robotics*, 22(5):890–902.
- Bosscher, P., Williams, R. L., Bryson, L. S., and Castro-Lacouture, D. (2007). Cable-suspended robotic contour crafting system. *Automation in Construction*, 17(1):45–55.

- Bouchard, S., Gosselin, C., and Moore, B. (2010). On the ability of a cable-driven robot to generate a prescribed set of wrenches. *Journal of Mechanisms and Robotics*, 2(1):011010.
- Carricato, M. (2013). Direct geometrico-static problem of underconstrained cable-driven parallel robots with three cables. *Journal of Mechanisms and Robotics*, 5(3):031008.
- Carricato, M. and Merlet, J.-P. (2010). Geometrico-static analysis of under-constrained cable-driven parallel robots. In Lenarčič, J. and Stanisic, M. M., editors, *Advances in Robot Kinematics: Motion in Man and Machine*, pages 309–319. Springer Netherlands.
- Carricato, M. and Merlet, J.-P. (2011). Direct geometrico-static problem of under-constrained cable-driven parallel robots with three cables. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 3011–3017.
- Carricato, M. and Merlet, J.-P. (2013). Stability analysis of underconstrained cable-driven parallel robots. *IEEE Transactions on Robotics*, 29(1):288–296.
- Collard, J.-F. and Cardou, P. (2013). Computing the lowest equilibrium pose of a cable-suspended rigid body. *Optimization and Engineering*, 14(3):457–476.
- Fattah, A. and Agrawal, S. K. (2005). On the design of cable-suspended planar parallel robots. *Journal of Mechanical Design*, 127(5):1021–1028.
- Fink, J., Michael, N., Kim, S., and Kumar, V. (2011). Planning and control for cooperative manipulation and transportation with aerial robots. *The International Journal of Robotics Research*, 30(3):324–334.
- Gao, B., Xu, J., Zhao, J., and Xi, N. (2012). Combined inverse kinematic and static analysis and optimal design of a cable-driven mechanism with a spring spine. *Advanced Robotics*, 26(8-9):923–946.
- Gouttefarde, M., Daney, D., and Merlet, J.-P. (2011). Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots. *IEEE Transactions on Robotics*, 27(1):1–13.
- Hansen, E. and Walster, G. W. (2003). *Global optimization using interval analysis: revised and expanded*. CRC Press.
- Heyden, T. and Woernle, C. (2006). Dynamics and flatness-based control of a kinematically undetermined cable suspension manipulator. *Multibody System Dynamics*, 16(2):155–177.
- Hiller, M., Fang, S., Mielczarek, S., Verhoeven, R., and Franitza, D. (2005). Design, analysis and realization of tendon-based parallel manipulators. *Mechanism and Machine Theory*, 40(4):429–445.
- Jaulin, L. (2001). *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer.
- Jiang, Q. and Kumar, V. (2010). The direct kinematics of objects suspended from cables. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Quebec, Canada, 15-18 August, 2010*, pages 193–202.
- Kawamura, S., Kino, H., and Won, C. (2000). High-speed manipulation by using parallel wire-driven robots. *Robotica*, 18(01):13–21.
- Kossowski, C. and Notash, L. (2002). Cat4 (cable actuated truss—4 degrees of freedom): A novel 4 dof cable actuated parallel manipulator. *Journal of Robotic Systems*, 19(12):605–615.
- Kozak, K., Zhou, Q., and Wang, J. (2006). Static analysis of cable-driven manipulators with non-negligible cable mass. *IEEE Transactions on Robotics*, 22(3):425–433.
- Merlet, J.-P. (2004a). Analysis of the influence of wires interference on the workspace of wire robots. In Lenarčič, J. and Galletti, C., editors, *On Advances in Robot Kinematics*, pages 211–218. Springer Netherlands.
- Merlet, J.-P. (2004b). Solving the forward kinematics of a gough-type parallel manipulator with interval analysis. *The International Journal of robotics research*, 23(3):221–235.
- Merlet, J.-P. (2007). Alias-c++. Available on line at: <http://www-sop.inria.fr/coprin/logiciels/ALIAS/ALIAS-C++/ALIAS-C++.html>.
- Merlet, J.-P. (2009). Interval analysis for certified numerical solution of problems in robotics. *International Journal of Applied Mathematics and Computer Science*, 19(3):399–412.
- Merlet, J.-P. (2010). Marionet, a family of modular wire-driven parallel robots. In Lenarčič, J. and Stanisic, M. M., editors, *Advances in Robot Kinematics: Motion in Man and Machine*, pages 53–61. Springer Netherlands.
- Merlet, J.-P. (2012). The kinematics of the redundant n–1 wire driven parallel robot. In *2012 IEEE International Conference on Robotics*

- and Automation, St. Paul, MN, USA, 14-18 May 2012, pages 2313–2318.
- Merlet, J.-P. and Alexandre-dit Sandretto, J. (2015). The forward kinematics of cable-driven parallel robots with sagging cables. In Pott, A. and Bruckmann, T., editors, *Cable-Driven Parallel Robots*, volume 32 of *Mechanisms and Machine Science*, pages 3–15. Springer International Publishing.
- Merlet, J.-P. and Daney, D. (2010). A portable, modular parallel wire crane for rescue operations. In *IEEE International Conference on Robotics and Automation, Anchorage, Alaska, 3-8 May 2010*, pages 2834–2839.
- Ming, A. and Higuchi, T. (1994a). Study on multiple degree-of-freedom positioning mechanism using wires. I: Concept, design and control. *International Journal of the Japan Society for Precision Engineering*, 28(2):131–138.
- Ming, A. and Higuchi, T. (1994b). Study on multiple degree-of-freedom positioning mechanism using wires. II: Development of a planar completely restrained positioning mechanism. *International Journal of the Japan Society for Precision Engineering*, 28(3):235–242.
- Moore, R. E. (1966). *Interval analysis*. Prentice-Hall Englewood Cliffs.
- Moore, R. E. (1979). *Methods and applications of interval analysis*. SIAM.
- Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.
- Pham, C. B., Yeo, S. H., Yang, G., Chen, I., et al. (2009). Workspace analysis of fully restrained cable-driven manipulators. *Robotics and Autonomous Systems*, 57(9):901–912.
- Pott, A., Mütterich, H., Kraus, W., Schmidt, V., Miermeister, P., and Verl, A. (2013). Ipanema: A family of cable-driven parallel robots for industrial applications. In Bruckmann, T. and Pott, A., editors, *Cable-Driven Parallel Robots*, volume 12 of *Mechanisms and Machine Science*, pages 119–134. Springer Berlin Heidelberg.
- Pusey, J., Fattah, A., Agrawal, S., and Messina, E. (2004). Design and workspace analysis of a 6–6 cable-suspended parallel robot. *Mechanism and machine theory*, 39(7):761–778.
- Ratschek, H. and Rokne, J. (1995). Interval methods. In *Handbook of global optimization*, pages 751–828. Springer.
- Roberts, R. G., Graham, T., and Lippitt, T. (1998). On the inverse kinematics, statics, and fault tolerance of cable-suspended robots. *Journal of Robotic Systems*, 15(10):581–597.
- Rohn, J. (1996). Enclosing solutions of overdetermined systems of linear interval equations. *Reliable Computing*, 2(2):167–171.
- Rosati, G., Gallina, P., and Masiero, S. (2007). Design, implementation and clinical tests of a wire-based robot for neurorehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4):560–569.
- Stump, E. and Kumar, V. (2006). Workspaces of cable-actuated parallel manipulators. *Journal of Mechanical Design*, 128(1):159–167.
- Su, Y., Duan, B., Nan, R., and Peng, B. (2001). Development of a large parallel-cable manipulator for the feed-supporting system of a next-generation large radio telescope. *Journal of Robotic Systems*, 18(11):633–643.
- Surdilovic, D., Zhang, J., and Bernhardt, R. (2007). String-man: Wire-robot technology for safe, flexible and human-friendly gait rehabilitation. In *IEEE 10th International Conference on Rehabilitation Robotics, Noordwijk, Netherlands, 13-15 June 2007*, pages 446–453.
- Tadokoro, S., Murao, Y., Hiller, M., Murata, R., Kohkawa, H., and Matsushima, T. (2002). A motion base with 6-dof by parallel cable drive architecture. *IEEE/ASME Transactions on Mechatronics*, 7(2):115–123.
- Tadokoro, S., Verhoeven, R., Hiller, M., and Takamori, T. (1999). A portable parallel manipulator for search and rescue at large-scale urban earthquakes and an identification algorithm for the installation in unstructured environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Kyongju, South Korea, 17–21 October 1999*, volume 2, pages 1222–1227 vol.2.
- Tapia, R. A. (1971). The kantorovich theorem for newton’s method. *American Mathematical Monthly*, pages 389–392.
- Yamamoto, M., Yanai, N., and Mohri, A. (2004). Trajectory control of incompletely restrained parallel-wire-suspended mechanism based on inverse dynamics. *IEEE Transactions on Robotics*, 20(5):840–850.