



**HAL**  
open science

# Interval-Analysis-Based Determination of the Wrench-Feasible Workspace of Parallel Cable-Driven Robots

Marc Gouttefarde, David Daney, Jean-Pierre Merlet

► **To cite this version:**

Marc Gouttefarde, David Daney, Jean-Pierre Merlet. Interval-Analysis-Based Determination of the Wrench-Feasible Workspace of Parallel Cable-Driven Robots. *IEEE Transactions on Robotics*, 2011, 27 (1), pp.001-013. 10.1109/TRO.2010.2090064 . lirmm-00573491

**HAL Id: lirmm-00573491**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00573491v1>**

Submitted on 29 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Interval Analysis Based Determination of the Wrench-Feasible Workspace of Parallel Cable-Driven Robots

Marc Gouttefarde, *Member, IEEE*, David Daney, *Member, IEEE*, and Jean-Pierre Merlet, *Member, IEEE*

**Abstract**—This paper deals with the wrench-feasible workspace (WFW) of  $n$ -degree-of-freedom parallel robots driven by  $n$  or more than  $n$  cables. The WFW is the set of mobile platform poses for which the cables can balance any wrench of a given set of wrenches, such that the tension in each cable remains within a prescribed range. Requirements of nonnegative cable tensions as well as maximum admissible tensions are thus satisfied. The determination of the WFW is an important issue since its size and shape are highly dependent on the geometry of the robot and on the ranges of allowed cable tensions. The approach proposed in this paper is mainly based on interval analysis. Two sufficient conditions are presented, namely, a sufficient condition for a box of poses to be fully inside the WFW and a sufficient condition for a box of poses to be fully outside the WFW. These sufficient conditions are relevant since they can be tested, means of testing them being discussed in the paper. Used within usual branch-and-prune algorithms, these tests enable WFW determinations in which full-dimensional sets of poses (volumes) are found to lie within or, on the contrary, to lie outside the WFW. This provides a useful alternative to a basic discretization, the latter consisting in testing a discrete (zero-dimensional) finite set of poses. In order to improve the efficiency of the computations, a means of mitigating the undesirable effects of the so-called wrapping effect is introduced. The paper also illustrates how the proposed approach is capable of dealing with small uncertainties on the geometric design parameters of a parallel cable-driven robot.

**Index Terms**—Cable-driven robots, workspace determination, wrench-feasibility, interval analysis.

## I. INTRODUCTION

PARALLEL cable-driven robots consist essentially of a mobile platform connected in parallel to actuated reels by cables as illustrated in Fig. 1. The actuation of each cable allows the control of the platform motion and/or wrenches. These parallel robots have several interesting properties such as a potentially very large reachable workspace since large lengths of cables can be used, low mass of the mobile parts of the robot and low visual intrusion. Therefore, they are good candidates for many applications, a few examples of which are

This work was supported in part by the French National Research Agency (ANR) under grant 2009-SEGI-018-01 and 2009-SEGI-018-02 and in part by the Région Languedoc-Roussillon under grant 115217.

Marc Gouttefarde is with the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Université Montpellier 2, UMR 5506 - CC 477, 161 rue Ada, 34095 Montpellier Cedex 5, France (e-mail: marc.gouttefarde@lirmm.fr).

David Daney and Jean-Pierre Merlet are with the COPRIN research team, INRIA Sophia Antipolis, 2004 route des lucioles, BP 93, 06902 Sophia Antipolis Cedex, France (e-mail: David.Daney@sophia.inria.fr; Jean-Pierre.Merlet@sophia.inria.fr).

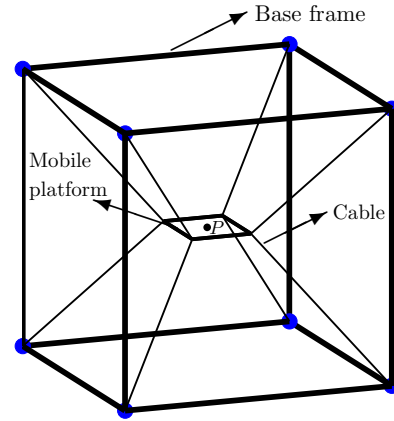


Fig. 1. A parallel cable-driven robot.

high-speed manipulation [1], [2], haptic interfaces [3], automated construction systems [4] and locomotion interfaces [5]. However, various factors may limit the workspace of parallel cable-driven robots among which the inability of the cables to push on the mobile platform is an important and challenging one. This fundamental issue has been the subject of several studies especially in recent years. The works presented in [6]–[13] are among the first ones to tackle this issue, and have been followed by several others. Cable interferences may also limit the workspace of parallel cable-driven robots. Cable collisions are generally avoided, *e.g.* [14], even if some authors suggest to permit collisions between the cables so as to obtain a larger workspace [15]. The present paper does not deal with collision issues. An interval analysis based method that can be used to check if cable interferences occur is presented in [16].

The consequences of the unilateral nature of cable actuation on the shape and size of the usable workspace of parallel cable-driven robots are usually examined by means of *wrench feasibility*. In general terms, a pose (position and orientation) of the mobile platform is said to be wrench feasible if any wrench of a given *required set of wrenches* can be applied to the mobile platform by pulling it with the cables (no cable has to push), possibly with a range of allowed tension values, that is a minimal (always nonnegative) and a maximal admissible value for the tension in each cable [17], [18]. The problem is then to determine the set of wrench feasible poses, *i.e.*, the so-called *wrench-feasible workspace* (WFW). Two particular instances of the WFW deserve their own names: the static (equilibrium)

workspace and the force-closure (FC) workspace, the latter being also known as the wrench-closure workspace—note that the terminology is far from being unified. The *static workspace* is the set of poses of the mobile platform for which the cables can balance the weight of both the platform and the payload with tension forces only. It is of particular interest for cable suspended robots such as the ROBOCRANE [8] which rely on gravity to keep the cables taut. The static workspace is a WFW whose required set of wrenches is reduced to a single wrench corresponding to the platform and payload weight. The *force-closure workspace*, whose name derives from an analogy with force-closure grasps of robotic hands [17], is the set of poses at which the platform can be fully constrained by the cables. The FC workspace is thus the very special case of a WFW whose required set of wrenches is the whole space of wrenches and the only constraint on the cable tensions is non-negativity.

Recently, “*geometric methods*” aiming at delineating the workspace boundary have been proposed in the case of the static [19], so-called dynamic [20] and FC [21] workspaces of planar robots. Not without difficulties, they have also been applied to the static and FC workspaces of six-DOF robots [22]. However, except for these particular instances of the WFW, geometric methods are not currently available probably due the number and complexity of the equations describing the WFW boundary [18]. Furthermore, because they aim at visualizing the boundary, these methods are limited to 3D workspaces. For six-DOF robots, only a partial picture of the workspace is thus obtained since at least three out of the six pose variables must be fixed, *e.g.*, when determining a constant-orientation workspace.

Therefore, the WFW is commonly determined by considering a grid of mobile platform poses, each pose of the grid being tested for wrench feasibility. The grid provides a discretization of the search space. In such a basic *discretization method*, the only tool needed is a test that determines if a given pose is wrench feasible. Classic methods of constrained optimization can usually be used to test wrench feasibility of a given pose. But, since feasibility rather than optimality is of interest, this does not necessarily yield very efficient tests. Therefore, more efficient approaches have been proposed. In the case of the FC workspace, references [10], [22]–[27] present specific tests which are based on characterizations of FC already known in grasping by robotic hands [28]. Related works can be found in [29], [30]. Only a few works, however, deals with the general case of the WFW, *i.e.*, with minimal and maximal admissible cable tensions. In fact, tests that determine if a given pose belongs to the WFW are discussed in [18], [31], [32], some of them focusing on a specific required wrench set geometry. The method proposed in [33] is a valuable one since it enables to test wrench feasibility of a given pose for a great variety of required wrench set geometries.

A discretization method is quite straightforward to implement but it does not truly solve the problem of determining the WFW. In fact, out of the infinitely many poses of the discretized search space, only finitely many of them—those of the discretization grid—are tested for wrench feasibility. Hence, all the poses of the grid may be wrench feasible while

the search space does not fully lie in the WFW because some poses which are not wrench feasible may be missed. Note that this issue is not unlikely to happen since the WFW has usually a non-convex geometry and can contains holes or consists of several disconnected components [21]. In this sense, *the result provided by a discretization can never be guaranteed, i.e.*, one can never know if this result can be trusted. In practice, a discretization should yield good results as soon as the discretization grid is fine enough. However, especially for six-DOF robots, the grid is to remain relatively coarse since otherwise the number of its poses is rapidly too large for all of them to be tested in a realistic amount of time. This implies a strong trade-off between accuracy (number of poses in the grid) and computation time. For instance, in the case of a 6D space of poses (a 6-DOF robot), when the discretization of each axis of the search space contains 10 points, provided that the discretization grid is defined as the Cartesian product of these discretizations, which is generally the case, the total number of poses to be tested is  $10^6$ , a million. With 25 points per axis, the total number of poses is greater than two billion and 100 points per axis leads to  $10^{12}$  poses, such that testing all of them is too expensive.

The present paper proposes an alternative to discretization with the aim of addressing the aforementioned drawbacks. This alternative consists of numerical methods, mainly based on interval analysis, providing guaranteed WFW determinations in reasonable computation times. It is applicable to any  $n$ -DOF parallel robot driven by  $n$  or more than  $n$  cables and consists mainly in a test able to find boxes of poses fully inside the WFW and a complementary test that can detect boxes fully outside the WFW. The determination is said to be guaranteed in the sense that, besides taking into account rounding errors, *wrench feasibility of each of the infinitely many poses of a search space is explored*. Possibly, the method yields sets of poses for which no conclusion could be drawn at the current algorithm resolution. The proposed approach is notably based on a theorem due to Rohn [34], which allows us to conclude on the feasibility of some infinitely many linear systems of equations by testing the feasibility of (only) finitely many of them (Section IV). It is also based on consistency techniques applied to continuous domains (Section V). Additionally, the proposed approach is able to deal with uncertainties on the geometric design parameters (in the form of small uncertainty boxes) in reasonable computation times, as long as these uncertainties remain small.

Some of the numerical tools proposed here have been introduced in a previous paper of the authors [35]. Besides, to the best of our knowledge, the only other work introducing interval analysis as a tool to compute the WFW is [36]. Compared to [36], the contributions of the present paper are:

- The use of Rohn’s Theorem (Section IV) to test if a box of poses is fully included in the WFW. This avoids both time consuming bisections in the cable tension and mobile platform wrench spaces and the needs of considering permutations of the wrench matrix columns.
- A means of testing whether a box of poses lies fully outside the WFW (Section V).
- A means of mitigating the effects of the so-called wrap-

ping effect (Section VII), thereby improving computation time.

These three points lead to interval analysis based determinations of the WFW having reasonable computation times.

The paper is organized as follows. The WFW is defined in Section II. Section III introduces interval evaluation of the wrench matrix of a parallel cable robot. Then, in Section IV, a detailed description of the main problem to be solved—testing the feasibility of infinitely many linear systems—is proposed. A sufficient condition for a box of poses to be fully inside the WFW is also introduced together with a means to test this condition. A sufficient condition for a box of poses to be fully outside the WFW and a means to test it are presented in Section V. In Section VI, the tests of Sections IV and V are used within usual branch-and-prune algorithms in order to determine the WFW. Then, in order to significantly improve efficiency, Section VII shows how to work with a wrench matrix without denominators. Finally, examples are provided in Section VIII and Section IX concludes the paper.

## II. WRENCH-FEASIBLE WORKSPACE DEFINITION

Let us consider an  $n$ -DOF parallel robot whose mobile platform is driven by  $m$  cables with  $m \geq n$ —generally  $n = 3$  or  $n = 6$ —together with a fixed frame  $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$  and a mobile frame  $(P, \mathbf{x}', \mathbf{y}', \mathbf{z}')$ , the latter being attached at the platform reference point  $P$  as shown in Fig. 2. The point of the fixed base from which cable  $i$  extends is denoted  $A_i$  whereas  $B_i$  denotes the point of the platform at which cable  $i$  is attached. If the projections of  $OA_i$ ,  $PB_i$  and  $OP$  in the base frame are denoted  $\mathbf{a}_i$ ,  $\mathbf{b}_i$  and  $\mathbf{p}$ , respectively, the vector directed along cable  $i$  from  $B_i$  to  $A_i$  is  $\mathbf{l}_i = \mathbf{a}_i - \mathbf{b}_i - \mathbf{p}$ . The corresponding unit vector is  $\mathbf{d}_i = \mathbf{l}_i / \rho_i$ , where  $\rho_i$  denotes the length of cable  $i$ . The pose of the mobile platform is defined by the position vector  $\mathbf{p}$  and by the orientation of the mobile frame with respect to the base frame.

The relationship between the tensions in the cables and the wrench  $\mathbf{f}$ —combination of a force and a moment—applied by the cables at the platform reference point  $P$  is [10], [37]

$$\mathbf{W}\boldsymbol{\tau} = \mathbf{f} \quad (1)$$

where  $\boldsymbol{\tau} = (\tau_1, \dots, \tau_m)^T$  is the vector of cable tensions and  $\mathbf{W}$  is the  $n \times m$  pose dependent *wrench matrix* whose columns are denoted  $\mathbf{w}_i$ ,  $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)$ . Thus, the wrench applied at  $P$  by cable  $i$  is  $\tau_i \mathbf{w}_i$ . In the case of six-DOF robots, the wrench  $\mathbf{w}_i$  is

$$\mathbf{w}_i = \begin{pmatrix} \mathbf{d}_i \\ \mathbf{b}_i \times \mathbf{d}_i \end{pmatrix}_{6 \times 1} \quad (2)$$

where  $\times$  denotes the cross product. Note that  $\mathbf{W} = -\mathbf{J}^{-T}$  where  $\mathbf{J}^{-1}$  is the so-called inverse Jacobian matrix [38] whose line vectors are normalized Plücker coordinates of the cable lines.

Let the *required wrench set*  $[\mathbf{f}]$  be the set of wrenches that the cables must apply at reference point  $P$  and let  $[\boldsymbol{\tau}]$  be the  $m$ -dimensional box of *allowed cable tensions*. Usually, the box  $[\boldsymbol{\tau}]$  is defined as

$$[\boldsymbol{\tau}] = \{\boldsymbol{\tau} \mid \tau_i \in [\tau_{i_{min}}, \tau_{i_{max}}], 1 \leq i \leq m\} \quad (3)$$

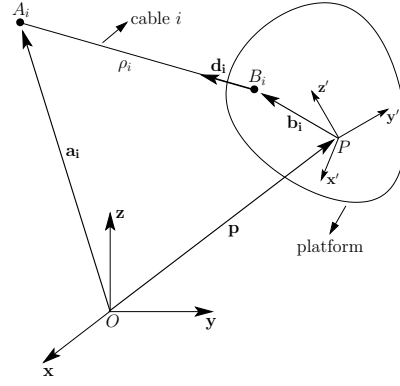


Fig. 2. Notations.

where, for each  $i$ ,  $\tau_{i_{min}}$  and  $\tau_{i_{max}}$  are two positive scalars such that  $\tau_{i_{min}} < \tau_{i_{max}}$ . A maximum tension  $\tau_{i_{max}}$  is necessary in order to take into account the maximum torque of each actuator or the maximum tension that a cable can withstand. The minimum tension  $\tau_{i_{min}} \geq 0$  allows one to ensure that none of the cables will be slack, a situation that may cause control problems.

In order to deal with the influence on the workspace of the unilateral nature of the forces applied by the cables, the WFW is of particular interest. It is defined as follows [18].

**Definition (WFW)** *The WFW is the set of mobile platform poses that are wrench feasible, i.e., for which, for any wrench  $\mathbf{f}$  in  $[\mathbf{f}]$ , there exists a vector of cable tensions  $\boldsymbol{\tau}$  in  $[\boldsymbol{\tau}]$  such that  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$ .*

Determining the WFW is useful since it allows to take into account both the requirement of nonnegative cable tensions and that of maximum admissible cable tensions. Additionally, at a wrench feasible pose, provided that  $[\mathbf{f}]$  contains a neighborhood of the origin, the wrench matrix  $\mathbf{W}$  has full rank. Generally, the WFW is thus a *singularity free workspace*.

## III. INTERVAL WRENCH MATRIX

### A. Interval Evaluation, Interval Vectors and Interval Matrices

This subsection provides a brief overview of interval analysis as used in the paper. More details can be found in [39]–[42].

An interval  $[x]$  is a bounded set of real numbers defined by

$$[x] = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \bar{x}\} \quad (4)$$

where  $(\underline{x}, \bar{x}) \in \mathbb{R}^2$  and  $\underline{x} \leq \bar{x}$ . The reals  $\underline{x}$  and  $\bar{x}$  are the lower and upper bound of the interval  $[x]$ , respectively. A fundamental feature of interval analysis is the *interval evaluation* of a function. Let us consider a real-valued function  $f(x)$ . The classic rules of addition, multiplication, etc. of real numbers can be extended to intervals yielding an interval arithmetic [39]. Moreover, by properly rounding the endpoints of computed intervals, rounding errors can be taken into account. Based on these basic operations and on the interval evaluation of basic algebraic and transcendental functions such as  $x^2$ ,  $\sin$  and  $\cos$ , almost any real function  $f(x)$  can be

evaluated for an interval  $[x]$  yielding an interval  $[f]$  which encloses the image of  $[x]$  under  $f$  (denoted  $f([x])$ , *i.e.*,

$$f([x]) = \{f(x) \mid x \in [x]\} \subseteq [f]. \quad (5)$$

The converse inclusion does not hold in general and  $[f]$  overestimates  $f([x])$  introducing pessimism in the evaluation. There exists several means to improve the interval evaluation of a function such as the use of the derivative of the function or of its Taylor series expansion, *e.g.* [39], [42].

An  $n$ -dimensional interval vector  $[\mathbf{x}] = ([x_1], \dots, [x_n])^T$  is a set of vectors  $\mathbf{x} = (x_1, \dots, x_n)^T$  such that, for each  $i$ ,  $x_i$  belongs to the interval  $[x_i]$ . For instance, the set of allowed cable tensions  $[\boldsymbol{\tau}]$  defined in (3) is an interval vector. Considered as a set of points of  $\mathbb{R}^n$ , an interval vector is a box shaped set. A real-valued function  $f(\mathbf{x})$  of several variables can be interval evaluated yielding an interval  $[f]$  that contains the image  $f([\mathbf{x}])$  of the interval vector  $[\mathbf{x}] = ([x_1], \dots, [x_n])^T$  under  $f$ , *i.e.*

$$f([\mathbf{x}]) = \{f(\mathbf{x}) \mid x_i \in [x_i], 1 \leq i \leq n\} \subseteq [f]. \quad (6)$$

Finally, an interval matrix  $[\mathbf{A}]$  is a two-dimensional rectangular array of intervals. We will denote  $[A_{ij}]$  the interval located at the  $i$ th line and  $j$ th column of  $[\mathbf{A}]$ . An interval matrix is a continuous set of matrices and the notation  $\mathbf{A} \in [\mathbf{A}]$  is understood component-wise, *i.e.*, by definition,  $\mathbf{A} \in [\mathbf{A}]$  if each real entry  $A_{ij}$  of matrix  $\mathbf{A}$  belongs to the corresponding interval  $[A_{ij}]$  of  $[\mathbf{A}]$ . By means of interval arithmetic, the multiplication of an interval matrix  $[\mathbf{A}]$  by an interval vector  $[\mathbf{x}]$  yields an interval vector  $[\mathbf{b}] = [\mathbf{A}][\mathbf{x}]$  such that

$$\forall \mathbf{A} \in [\mathbf{A}], \forall \mathbf{x} \in [\mathbf{x}], \mathbf{A}\mathbf{x} \in [\mathbf{b}]. \quad (7)$$

In the sense of (7),  $[\mathbf{b}]$  is a box enclosure of the set

$$\{\mathbf{A}\mathbf{x} \mid \mathbf{A} \in [\mathbf{A}] \text{ and } \mathbf{x} \in [\mathbf{x}]\} \quad (8)$$

which generally is not a box. Hence, there usually exist vectors  $\mathbf{b} \in [\mathbf{b}]$  which do not belong to this set, *i.e.*, such that

$$\forall \mathbf{A} \in [\mathbf{A}], \forall \mathbf{x} \in [\mathbf{x}], \mathbf{b} \neq \mathbf{A}\mathbf{x}. \quad (9)$$

### B. Interval Evaluation of the Wrench Matrix

Let the  $n$ -dimensional vector  $\mathbf{x}$  denote the pose of the mobile platform. For instance, in the case of a 6-DOF robot,  $\mathbf{x}$  can be the six-dimensional vector  $(\mathbf{p}^T, \boldsymbol{\psi}^T)^T$  where  $\mathbf{p}$  defines the position of the platform and the three components of vector  $\boldsymbol{\psi}$  are Euler angles defining its orientation.

Consider a box of mobile platform poses. This box is an  $n$ -dimensional interval vector, which we denote by  $[\mathbf{x}]$ . Each component  $w_{ij}$  of the wrench matrix  $\mathbf{W}$  is pose dependent. Such a dependency is denoted by  $w_{ij} = w_{ij}(\mathbf{x})$ . Referring to the preceding section, each  $w_{ij}$  can be interval evaluated for the interval vector  $[\mathbf{x}]$  yielding an interval  $[w_{ij}]$ . According to (6),  $[w_{ij}]$  satisfies the following property: for all poses  $\mathbf{x}$  in  $[\mathbf{x}]$ ,  $w_{ij}(\mathbf{x})$  belongs to  $[w_{ij}]$ . Therefore, the  $n \times m$  interval matrix  $[\mathbf{W}]$  whose components are the intervals  $[w_{ij}]$  has the following *fundamental property*

$$\forall \mathbf{x} \in [\mathbf{x}], \mathbf{W}(\mathbf{x}) \in [\mathbf{W}]. \quad (10)$$

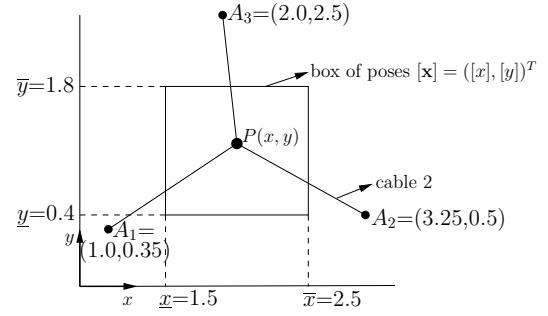


Fig. 3. A point-mass cable-driven robot and its box workspace  $[\mathbf{x}]$ .

In (10),  $\mathbf{W}(\mathbf{x})$  stands for the wrench matrix obtained for the pose  $\mathbf{x}$ . In words, for each and every pose  $\mathbf{x} \in [\mathbf{x}]$ , the wrench matrix  $\mathbf{W}$  obtained for  $\mathbf{x}$  belongs to  $[\mathbf{W}]$ . The interval matrix  $[\mathbf{W}]$  is referred to as the *interval wrench matrix*. Let us note here that (10) must be verified for the results presented in Section IV-A and V-A to hold. The interval wrench matrix can be thought of as a box in  $\mathbb{R}^{nm}$  which encloses the set

$$\{\mathbf{W}(\mathbf{x}) \mid \mathbf{x} \in [\mathbf{x}]\} \quad (11)$$

of the infinitely many wrench matrices corresponding to the poses  $\mathbf{x}$  lying in  $[\mathbf{x}]$ . The set (11) has an unknown complex shape since each component  $w_{ij}$  of  $\mathbf{W}$  is a nonlinear function of  $\mathbf{x}$ . Consequently, the interval wrench matrix “overestimates” (11) and it is important to note that there exist matrices  $\mathbf{W}_\emptyset \in [\mathbf{W}]$  which are not wrench matrices, *i.e.*,  $\forall \mathbf{x} \in [\mathbf{x}], \mathbf{W}_\emptyset \neq \mathbf{W}(\mathbf{x})$ . This is known as the *wrapping effect* [40] which refers to the loss of the dependency relationships between the  $w_{ij}$  at the time of the determination of  $[\mathbf{W}]$ . In the sequel, the notation  $[\mathbf{W}] = [\mathbf{W}]([\mathbf{x}])$  means that  $[\mathbf{W}]$  is the interval matrix obtained by interval evaluating each component of  $\mathbf{W}$  for the box  $[\mathbf{x}]$ . Hence, it verifies (10). In the pseudocodes presented in Section VI, this computation is denoted `Compute_WrenchMatrix`( $[\mathbf{x}]$ ).

Note that the wrench matrix  $\mathbf{W}$  is also a function of the geometric design parameters  $\mathbf{a}_i$  and  $\mathbf{b}_i$  defined in Section II. Let us collect all these design parameters in a vector  $\mathbf{u}$ , *e.g.*  $\mathbf{u} = (a_{1x}, a_{1y}, a_{1z}, \dots, b_{mx}, b_{my}, b_{mz})$ . If in addition to a box  $[\mathbf{x}]$ , a box  $[\mathbf{u}]$  of design parameters is given, the wrench matrix  $\mathbf{W}$  can be interval evaluated for  $[\mathbf{x}]$  and  $[\mathbf{u}]$  yielding an interval wrench matrix  $[\mathbf{W}]$  such that  $\forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{u} \in [\mathbf{u}], \mathbf{W}(\mathbf{x}, \mathbf{u}) \in [\mathbf{W}]$ .

For instance, let us consider a simple point-mass two-DOF robot driven by three cables as shown in Fig. 3. The wrench matrix of this robot is the  $2 \times 3$  matrix  $\mathbf{W} = (\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$  where  $\mathbf{d}_i$  is the two-dimensional unit vector directed along cable  $i$  from  $P$  to  $A_i$ . For the box of poses  $[\mathbf{x}] = ([1.5, 2.5], [0.4, 1.8])^T$ , each component of each  $\mathbf{d}_i$  is interval evaluated yielding the following  $[\mathbf{W}]$

$$\begin{pmatrix} [-2.986, -0.239] & [0.344, 2.334] & [-0.715, 0.715] \\ [-2.886, -0.0239] & [-1.734, 0.134] & [0.324, 3.0] \end{pmatrix}. \quad (12)$$

It can be verified that for *any* position  $\mathbf{x}$  of  $P$  lying in  $[\mathbf{x}]$ , the associated wrench matrix  $\mathbf{W}$  lies within  $[\mathbf{W}]$ .

Finally, note that, for instance, although the matrix

$$\begin{pmatrix} -1 & 1.5 & 0 \\ -0.5 & 0 & 1.6 \end{pmatrix} \quad (13)$$

belongs to  $[\mathbf{W}]$ , it does not correspond to any wrench matrix obtained for  $\mathbf{x}$  in  $[\mathbf{x}]$ .

#### IV. BOXES FULLY INCLUDED IN THE WFW

##### A. Feasibility of Interval Linear Systems

In this paper, *the WFW considered are such that the required wrench set has the shape of a box in  $\mathbb{R}^n$*  and, thus, can be identified with an  $n$ -dimensional interval vector  $[\mathbf{f}]$ .

In order to compute the WFW by means of branch-and-prune algorithms (Section VI), one of the main problem consists in verifying whether a given box (interval vector) of poses  $[\mathbf{x}]$  is fully included in this workspace. According to the definition of the WFW, this problem amounts to testing if

$$\forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{f} \in [\mathbf{f}], \exists \boldsymbol{\tau} \in [\boldsymbol{\tau}] \text{ such that } \mathbf{W}\boldsymbol{\tau} = \mathbf{f}. \quad (14)$$

In words, (14) reads that for any pose  $\mathbf{x}$  in  $[\mathbf{x}]$  and for any wrench  $\mathbf{f}$  in  $[\mathbf{f}]$  to be exerted by the cables on the mobile platform, there exists cable tensions  $\boldsymbol{\tau}$  in the allowed set  $[\boldsymbol{\tau}]$  that can generate  $\mathbf{f}$ , *i.e.*, such that  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$ .

Let  $\mathbf{x} \in [\mathbf{x}]$  be a pose of the mobile platform and  $\mathbf{W}$  the corresponding wrench matrix. For a given wrench  $\mathbf{f} \in [\mathbf{f}]$ , the system of linear equations  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$  is said to be *feasible* if it admits a solution  $\boldsymbol{\tau}$  in  $[\boldsymbol{\tau}]$ . Then, since there are infinitely many poses  $\mathbf{x}$  in  $[\mathbf{x}]$  and infinitely many wrenches  $\mathbf{f}$  in  $[\mathbf{f}]$ , *verifying whether (14) is true amounts to verifying the feasibility of infinitely many linear systems  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$* . In other words, it requires the feasibility of all the individual systems belonging to the following continuous family of linear systems

$$\{\mathbf{W}\boldsymbol{\tau} = \mathbf{f} \mid \mathbf{x} \in [\mathbf{x}], \mathbf{f} \in [\mathbf{f}]\}. \quad (15)$$

To solve this difficult problem, one can rely on interval analysis tools that allow to deal with a so-called *system of interval linear equations* [41], [42], denoted

$$[\mathbf{W}]\boldsymbol{\tau} = [\mathbf{f}] \quad (16)$$

which, following [34], is said to be *strongly feasible* when

$$\forall \mathbf{W} \in [\mathbf{W}], \forall \mathbf{f} \in [\mathbf{f}], \exists \boldsymbol{\tau} \in [\boldsymbol{\tau}] \text{ such that } \mathbf{W}\boldsymbol{\tau} = \mathbf{f}. \quad (17)$$

In fact, let us consider a given box of poses  $[\mathbf{x}]$  and the associated wrench matrix  $[\mathbf{W}] = [\mathbf{W}]([\mathbf{x}])$ . According to (10), (17) implies (14). Hence, *the strong feasibility (17) of the system of interval linear equations (16) is a sufficient condition for the box  $[\mathbf{x}]$  to be fully included in the WFW*. Nevertheless, since the interval wrench matrix  $[\mathbf{W}]$  is only an enclosure and not an exact description of the set (11), (17) *is sufficient but not necessary for  $[\mathbf{x}]$  to be fully inside the WFW*.

A theorem, due to Rohn, providing a necessary and sufficient condition for (17) to be true is presented in Section IV-C. This theorem applies to the case  $m \geq n$  of  $n$ -DOF parallel robots driven by  $n$  or more than  $n$  cables. In the particular case  $m = n$  (as many cables as DOF), note that well-known tools of interval analysis dealing with square systems of interval linear equations [41], [42] can also be applied.

##### B. A Set of Vertex Matrices and Vertex Vectors

Before introducing Rohn's Theorem in the next subsection, a set of "vertex" matrices of an interval matrix has to be defined. Let  $[\mathbf{A}]$  be an  $n \times m$  interval matrix whose components are the intervals  $[A_{ij}] = [\underline{A}_{ij}, \overline{A}_{ij}]$  and let  $Y_n$  be the set of  $n$ -dimensional vectors  $\mathbf{y}$  whose components  $y_i$  are equal either to 1 or to -1. We associate with each of the  $2^n$  vectors  $\mathbf{y} \in Y_n$  the matrix  $\mathbf{A}_{\mathbf{y}}$  whose components are

$$A_{\mathbf{y}_{ij}} = \underline{A}_{ij} + (\overline{A}_{ij} - \underline{A}_{ij})(1 - y_i)/2. \quad (18)$$

Note that the  $i$ th component  $y_i$  of  $\mathbf{y}$  defines completely the  $i$ th line of  $\mathbf{A}_{\mathbf{y}}$ , *e.g.*, when  $y_i = -1$  the components of the  $i$ th line of  $\mathbf{A}_{\mathbf{y}}$  are equal to the upper bounds  $\overline{A}_{ij}$  of the corresponding interval components of the  $i$ th line of  $[\mathbf{A}]$ . The  $2^n$  matrices  $\mathbf{A}_{\mathbf{y}}$  are called vertex matrices since, as the interval matrix  $[\mathbf{A}]$  can be thought of as a box embedded in  $\mathbb{R}^{nm}$ , the  $\mathbf{A}_{\mathbf{y}}$  corresponds to some (but not all) of the vertices of this box. For instance, with  $\mathbf{y} = (1, -1)^T \in Y_2$ , the vertex matrix  $\mathbf{W}_{\mathbf{y}}$  of the interval matrix  $[\mathbf{W}]$  defined in (12) is equal to

$$\begin{pmatrix} -2.986 & 0.344 & -0.715 \\ -0.0239 & 0.134 & 3.0 \end{pmatrix}. \quad (19)$$

Let us also define the set of "vertex" vectors of an interval vector  $[\mathbf{b}]$  as the set of the  $2^n$  vectors  $\mathbf{b}_{\mathbf{y}}$ ,  $\mathbf{y}$  in  $Y_n$ , whose components are defined by

$$b_{\mathbf{y}_i} = \underline{b}_i + (\overline{b}_i - \underline{b}_i)(1 + y_i)/2. \quad (20)$$

where the interval  $[\underline{b}_i, \overline{b}_i]$  is the  $i$ th component of  $[\mathbf{b}]$ . Geometrically, the set of vectors  $\mathbf{b}_{\mathbf{y}}$  is the set of vertices of the  $n$ -dimensional box  $[\mathbf{b}]$ .

##### C. A Useful Theorem

The system of interval linear equations (16) is a shorthand notation for the set of linear systems

$$\{\mathbf{W}\boldsymbol{\tau} = \mathbf{f} \mid \mathbf{W} \in [\mathbf{W}] \text{ and } \mathbf{f} \in [\mathbf{f}]\}. \quad (21)$$

The following theorem states that the *infinitely* many systems belonging to (21) are all feasible, *i.e.*, (17) is true, if and only if *finitely* many of them are feasible.

**Theorem 1 (Rohn)** *The system of interval linear equations  $[\mathbf{W}]\boldsymbol{\tau} = [\mathbf{f}]$  is strongly feasible if and only if the  $2^n$  systems of linear equations  $\mathbf{W}_{\mathbf{y}}\boldsymbol{\tau} = \mathbf{f}_{\mathbf{y}}$ ,  $\mathbf{y} \in Y_n$ , are all feasible.*

The necessary condition is trivial whereas the reader interested in the proof of the sufficient condition is referred to [34] (Chapter 2). Since strong feasibility (17) is a sufficient condition for a box  $[\mathbf{x}]$  to lie within the WFW, this very useful theorem allows us to conclude that  $[\mathbf{x}]$  is fully inside the WFW whenever each of the  $2^n$  (classic) linear systems  $\mathbf{W}_{\mathbf{y}}\boldsymbol{\tau} = \mathbf{f}_{\mathbf{y}}$ ,  $\mathbf{y} \in Y_n$ , is feasible.  $\mathbf{W}_{\mathbf{y}}$  and  $\mathbf{f}_{\mathbf{y}}$  are the vertex matrix of  $[\mathbf{W}]$  and the vertex vector of  $[\mathbf{f}]$  defined by (18) and (20), respectively.

Note that the feasibility of a system  $\mathbf{W}_{\mathbf{y}}\boldsymbol{\tau} = \mathbf{f}_{\mathbf{y}}$  is a well-known problem in linear programming (LP). For example, it can be tested by means of the simplex method applied to the general LP problem [43] (chapter 8)

$$\min \mathbf{c}^T \boldsymbol{\tau} \quad \text{s.t.} \quad \mathbf{W}_{\mathbf{y}}\boldsymbol{\tau} = \mathbf{f}_{\mathbf{y}} \text{ and } \boldsymbol{\tau} \in [\boldsymbol{\tau}] \quad (22)$$

for which any linear form  $\mathbf{c}$ , e.g. the trivial linear form  $\mathbf{c} = \mathbf{0}$ , can be considered since only the feasibility of the problem is to be determined. Alternatively, one can also rely on the methods presented in [18], [31]–[33]. Note that the method introduced in [44] allows one to keep the numerical guarantee of interval analysis (the simplex method being highly sensitive to rounding errors).

In brief, by means of Theorem 1, a procedure denoted  $\text{Feasible}([\mathbf{W}], [\mathbf{f}], [\boldsymbol{\tau}])$  can be written. This procedure returns **true** if the necessary and sufficient condition stated in Theorem 1 is true and it returns **unknown** otherwise. When  $\text{Feasible}()$  returns true, it thus ensures that the box of poses  $[\mathbf{x}]$  for which  $[\mathbf{W}]$  has been calculated is fully included in the WFW.  $\text{Feasible}()$  consists in testing the feasibility of at most  $2^n$  linear systems  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$ .

## V. BOXES FULLY OUTSIDE THE WFW

A means of testing whether a given box of poses  $[\mathbf{x}]$  lies fully outside the WFW is a valuable tool. Indeed, when  $\text{Feasible}()$  returns **true**, the box  $[\mathbf{x}]$  at hand is guaranteed to be fully included in the WFW. But, when  $\text{Feasible}()$  returns **unknown** nothing is known about the location of  $[\mathbf{x}]$  with respect to the WFW since  $\text{Feasible}()$  is based on condition (17), the latter being only a sufficient condition for  $[\mathbf{x}]$  to be fully inside the WFW.

### A. A Sufficient Condition

By definition, a pose  $\mathbf{x}$  of the mobile platform of an  $n$ -DOF parallel robot driven by  $m$  cables belongs to the WFW if

$$\forall \mathbf{f} \in [\mathbf{f}], \exists \boldsymbol{\tau} \in [\boldsymbol{\tau}] \text{ such that } \mathbf{W}\boldsymbol{\tau} = \mathbf{f}. \quad (23)$$

A direct consequence of the convexity of  $[\mathbf{f}]$  and of the set  $\mathcal{F} = \{\mathbf{f} \mid \mathbf{f} = \mathbf{W}\boldsymbol{\tau}, \boldsymbol{\tau} \in [\boldsymbol{\tau}]\}$ —the available net wrench set [18]—is the equivalence between (23) and the following condition

$$\forall \mathbf{f}_y \in \{\mathbf{f}_y\}, \exists \boldsymbol{\tau} \in [\boldsymbol{\tau}] \text{ such that } \mathbf{W}\boldsymbol{\tau} = \mathbf{f}_y \quad (24)$$

in which, using the notations of Section IV-B,  $\{\mathbf{f}_y\}$  denotes the set of vertices of the box  $[\mathbf{f}]$ . A proof of this equivalence can be found in [18] in the case  $\tau_{i_{\min}} = 0$ .

A single pose  $\mathbf{x}$  lies outside the WFW if and only if (24) is false. Hence, a box of poses  $[\mathbf{x}]$  is fully outside the WFW if and only if (24) is false for any  $\mathbf{x} \in [\mathbf{x}]$ , that is, if and only if

$$\forall \mathbf{x} \in [\mathbf{x}], \exists \mathbf{f}_y \in \{\mathbf{f}_y\} \text{ such that } \forall \boldsymbol{\tau} \in [\boldsymbol{\tau}], \mathbf{W}\boldsymbol{\tau} \neq \mathbf{f}_y. \quad (25)$$

In words,  $[\mathbf{x}]$  is fully outside the WFW if and only if, for any pose  $\mathbf{x}$  in  $[\mathbf{x}]$ , there exists at least one vertex  $\mathbf{f}_y$  of the required wrench set  $[\mathbf{f}]$  which cannot be generated by the cables with tensions in  $[\boldsymbol{\tau}]$ . Equation (25) is a necessary and sufficient condition for a box of poses  $[\mathbf{x}]$  to be fully outside the WFW but it appears to be difficult to test it. Instead, we propose to consider the following condition

$$\exists \mathbf{f}_y \in \{\mathbf{f}_y\} \text{ such that } \forall \mathbf{W} \in [\mathbf{W}], \forall \boldsymbol{\tau} \in [\boldsymbol{\tau}], \mathbf{W}\boldsymbol{\tau} \neq \mathbf{f}_y \quad (26)$$

where  $[\mathbf{W}] = [\mathbf{W}]([\mathbf{x}])$ . Since the implication (26)  $\Rightarrow$  (25) holds, (26) is a sufficient condition for a box  $[\mathbf{x}]$  to be fully

outside the WFW. The converse implication is generally not true so that *this condition is not necessary*. The implication (26)  $\Rightarrow$  (25) is a consequence of (10) and of the fact that in (25) the wrench  $\mathbf{f}_y$  which cannot be generated with admissible tensions  $\boldsymbol{\tau}$  depends upon the pose  $\mathbf{x}$  considered in  $[\mathbf{x}]$  whereas, in (26),  $\mathbf{f}_y$  does not depend on  $\mathbf{x}$ , i.e., it cannot be generated with admissible cable tensions whatever  $\mathbf{W} \in [\mathbf{W}]$  and thus whatever  $\mathbf{x} \in [\mathbf{x}]$ . In this sense, (26) is stronger than (25).

### B. A Test Based on Consistency

For a given interval wrench matrix  $[\mathbf{W}]$ , (26) can be tested as follows: the vertices  $\mathbf{f}_y$  of  $[\mathbf{f}]$  are considered in turn and, for each  $\mathbf{f}_y$ , we check if the system of interval linear equations  $[\mathbf{W}]\boldsymbol{\tau} = \mathbf{f}_y$  is inconsistent, that is, whether or not

$$\forall \mathbf{W} \in [\mathbf{W}], \forall \boldsymbol{\tau} \in [\boldsymbol{\tau}], \mathbf{W}\boldsymbol{\tau} \neq \mathbf{f}_y. \quad (27)$$

To this end, one can rely on so-called *consistency techniques* applied to continuous domains [42], [45], [46]. These numerical techniques do not aim at solving the system of equations but rather at narrowing down the size of the box  $[\boldsymbol{\tau}]$  such that no solution is lost. Two examples are hull consistency and 3B-consistency which can be found in [47], for example.

Due to space limitations, details on consistency techniques applied to continuous domains are not provided in this paper. The important point for the problem at hand is that applying a consistency technique to the system  $[\mathbf{W}]\boldsymbol{\tau} = \mathbf{f}_y$  with variables  $\boldsymbol{\tau}$  and associated domain  $[\boldsymbol{\tau}]$  yields a new box  $[\boldsymbol{\tau}]'$  such that, on the one hand,  $[\boldsymbol{\tau}]' \subseteq [\boldsymbol{\tau}]$  and, on the other hand, if

$$\exists \mathbf{W} \in [\mathbf{W}], \exists \boldsymbol{\tau} \in [\boldsymbol{\tau}] \text{ such that } \mathbf{W}\boldsymbol{\tau} = \mathbf{f}_y \quad (28)$$

then  $\boldsymbol{\tau} \in [\boldsymbol{\tau}]'$ , i.e., no solution is lost. Moreover, if the consistency technique returns  $[\boldsymbol{\tau}]' = \{\emptyset\}$  (where  $\{\emptyset\}$  denotes the empty set), the system of interval linear equations  $[\mathbf{W}]\boldsymbol{\tau} = \mathbf{f}_y$  with domain  $[\boldsymbol{\tau}]$  is inconsistent, i.e., (27) is true. In summary, if for one vertex  $\mathbf{f}_y$  of  $[\mathbf{f}]$ , a consistency technique applied to  $[\mathbf{W}]\boldsymbol{\tau} = \mathbf{f}_y$  yields a new domain  $[\boldsymbol{\tau}]'$  such that  $[\boldsymbol{\tau}]' = \{\emptyset\}$  then the box of poses  $[\mathbf{x}]$  is fully outside the WFW. Thus, (26) together with a consistency technique provides a procedure denoted  $\text{Out}([\mathbf{W}], \{\mathbf{f}_y\}, [\boldsymbol{\tau}])$  that considers the vertices  $\mathbf{f}_y$  of  $[\mathbf{f}]$  in turn. If for one of these vertices the consistency technique changes  $[\boldsymbol{\tau}]$  so that it becomes the empty set, (27) is verified, the current box of poses  $[\mathbf{x}]$  is completely outside the WFW and the procedure  $\text{Out}()$  returns **true**. Otherwise,  $\text{Out}()$  returns **unknown**.

## VI. DETERMINATION OF THE WFW: BRANCH-AND-PRUNE ALGORITHMS

This section presents examples of pseudocodes that illustrate how the previously introduced procedures  $\text{Feasible}()$  and  $\text{Out}()$  can be used to determine the WFW by means of  $n$ -dimensional boxes, to test if a given prescribed workspace is fully included in the WFW and to obtain the total orientation WFW. These pseudocodes are essentially branch-and-prune algorithms.

### A. Determination of the WFW

The algorithm shown in Fig. 4 manages a list  $\mathcal{L}$  of pose boxes  $[\mathbf{x}]$  initialized with the search box  $\mathcal{B}$ . The current box  $[\mathbf{x}]$  is found to be either fully inside the WFW (line 5), or fully outside the WFW (line 7), or else  $[\mathbf{x}]$  is bisected (line 11) as long as it is not too small according to the threshold  $\epsilon$ . In the latter case, the two resulting smaller boxes are put back in the list  $\mathcal{L}$  so that each becomes the current box at a later time during the execution of the algorithm.

This algorithm outputs the lists  $\mathcal{L}_{in}$ ,  $\mathcal{L}_{out}$  and  $\mathcal{L}_{neg}$ . The properties of the procedure `Feasible()` (resp. `Out()`) ensure that all the boxes of  $\mathcal{L}_{in}$  (resp.  $\mathcal{L}_{out}$ ) are fully included in the WFW (resp. fully outside). The boxes of  $\mathcal{L}_{neg}$  are of two types.

- A box  $[\mathbf{x}]$  of the *first type* is a box too small to be bisected and containing a part of the WFW boundary.
- A box  $[\mathbf{x}]$  of the *second type* is too small to be bisected but does not contain any WFW boundary pose: it is either fully inside or else fully outside the WFW but not detected as such at line 5 or 7. Such boxes exist because the procedures `Feasible()` and `Out()` are based on conditions which are only sufficient (Sections IV and V). This is a negative consequence of the wrapping effect, which necessarily comes along with the computation of  $[\mathbf{W}]$  (Section III-B), and of the fact that the problem of finding boxes fully outside the WFW has been relaxed so as to obtain the procedure `Out()` (condition (26) is stronger than (25), see Section V).

Due to the existence of boxes  $[\mathbf{x}]$  of the second type,  $\mathcal{L}_{neg}$  provides a “*thick*” representation of the WFW boundary.

The accuracy of the algorithm, *i.e.* the size of the boxes of  $\mathcal{L}_{neg}$ , can be adjusted by means of the threshold  $\epsilon$ . Obviously, the smaller is  $\epsilon$  the higher the computation time.

The WFW boundary is always contained within  $\mathcal{L}_{neg}$ , *i.e.*, the boxes of  $\mathcal{L}_{neg}$  provides a *conservative approximation of the WFW boundary*. Indeed, consider a pose  $\mathbf{x}$  of  $\mathcal{B}$  which lies on the WFW boundary and let us show that  $\mathbf{x}$  belongs necessarily to  $\mathcal{L}_{neg}$ . Assume to the contrary that  $\mathbf{x}$  belongs only to boxes of  $\mathcal{L}_{in}$  and/or of  $\mathcal{L}_{out}$ . The WFW being a closed set,  $\mathbf{x} \in \text{WFW}$  and thus  $\mathbf{x}$  cannot belong to  $\mathcal{L}_{out}$ . Hence,  $\mathbf{x}$  belongs solely to boxes of  $\mathcal{L}_{in}$ . Then, either  $\mathbf{x}$  is an interior point of a (unique) box of  $\mathcal{L}_{in}$  or  $\mathbf{x}$  lies on the common boundary of several boxes of  $\mathcal{L}_{in}$ . But, in both cases, there exists a neighborhood of  $\mathbf{x}$  lying in the union of the boxes of  $\mathcal{L}_{in}$  containing  $\mathbf{x}$  and consequently fully included in the WFW. Then,  $\mathbf{x}$  lies in the WFW interior in contradiction with the fact that it is on the WFW boundary. In conclusion, a pose  $\mathbf{x}$  of  $\mathcal{B}$  lying on the WFW boundary belongs neither to  $\mathcal{L}_{in}$  nor to  $\mathcal{L}_{out}$  and thus necessarily belongs to (a box of)  $\mathcal{L}_{neg}$ . This shows that the part of the WFW boundary lying in the search box  $\mathcal{B}$  is necessarily contained in boxes of  $\mathcal{L}_{neg}$ .

Finally, note that the structure of the algorithm shown in Fig. 4—as well as those of the algorithms described in the two following subsections—makes it well adapted to parallel computing.

**Input:**  $\mathcal{B}$ ,  $[\mathbf{f}]$ ,  $[\boldsymbol{\tau}]$ ,  $\epsilon$

**Output:**  $\mathcal{L}_{in}$ ,  $\mathcal{L}_{out}$ ,  $\mathcal{L}_{neg}$

```

1.  $\mathcal{L} \leftarrow \mathcal{B}$  % Initialize list  $\mathcal{L}$  with the search box  $\mathcal{B}$ 
2. while  $\mathcal{L} \neq \{\emptyset\}$  do
3.    $[\mathbf{x}] \leftarrow \text{Extract}(\mathcal{L})$ 
4.    $[\mathbf{W}] = \text{Compute\_WrenchMatrix}([\mathbf{x}])$ 
5.   if Feasible( $[\mathbf{W}]$ ,  $[\mathbf{f}]$ ,  $[\boldsymbol{\tau}]$ ) then
6.      $\mathcal{L}_{in} \leftarrow [\mathbf{x}]$  %  $[\mathbf{x}] \subseteq \text{WFW}$ 
7.   else if Out( $[\mathbf{W}]$ ,  $\{\mathbf{f}_y\}$ ,  $[\boldsymbol{\tau}]$ ) then
8.      $\mathcal{L}_{out} \leftarrow [\mathbf{x}]$  %  $[\mathbf{x}] \cap \text{WFW} = \{\emptyset\}$ 
9.   else % Both Feasible() and Out() returned unknown
10.    if Size( $[\mathbf{x}]$ )  $> \epsilon$  then
11.       $\mathcal{L} \leftarrow \text{Bisect}([\mathbf{x}])$ 
12.    else %  $[\mathbf{x}]$  is too small to be bisected
13.       $\mathcal{L}_{neg} \leftarrow [\mathbf{x}]$ 
14.    end if
15.  end if
16. end while

```

Fig. 4. A branch-and-prune algorithm to determine the WFW.

**Input:**  $\mathcal{W}$ ,  $[\mathbf{f}]$ ,  $[\boldsymbol{\tau}]$ ,  $\mathcal{L}$ ,  $\epsilon$

**Output:**  $\mathcal{L}$

```

1. if  $\mathcal{L} = \{\emptyset\}$  then
2.    $\mathcal{L} \leftarrow \mathcal{W}$  % Initialize list  $\mathcal{L}$  with the box workspace  $\mathcal{W}$ 
3. end if
4. while  $\mathcal{L} \neq \{\emptyset\}$  do
5.    $[\mathbf{x}] \leftarrow \text{Extract}(\mathcal{L})$ 
6.    $[\mathbf{W}] = \text{Compute\_WrenchMatrix}([\mathbf{x}])$ 
7.   if Out( $[\mathbf{W}]$ ,  $\{\mathbf{f}_y\}$ ,  $[\boldsymbol{\tau}]$ ) then
8.     return false
9.   else if Feasible( $[\mathbf{W}]$ ,  $[\mathbf{f}]$ ,  $[\boldsymbol{\tau}]$ ) then
10.    continue %  $[\mathbf{x}] \subseteq \text{WFW}$ 
11.   else
12.    if Size( $[\mathbf{x}]$ )  $> \epsilon$  then
13.       $\mathcal{L} \leftarrow \text{Bisect}([\mathbf{x}])$ 
14.    else %  $[\mathbf{x}]$  is too small to be bisected
15.       $\mathcal{L} \leftarrow [\mathbf{x}]$ 
16.    return unknown
17.   end if
18.   end if
19. end while
20. return true

```

Fig. 5. Testing a prescribed workspace  $\mathcal{W}$  for full inclusion in the WFW.

### B. Test of a Box Workspace $\mathcal{W}$

Fig. 5 shows the pseudocode of an algorithm using the same ingredients as those of Fig. 4 in order to test whether or not a given prescribed workspace  $\mathcal{W}$  (a big box) is fully included in the WFW. When this algorithm returns **true**, it ensures that  $\mathcal{W}$  is fully included in the WFW since all of the infinitely many poses of  $\mathcal{W}$  have been checked for wrench feasibility. When **false** is returned, a part of  $\mathcal{W}$  lies completely outside the WFW (line 8). Finally, when the state **unknown** is returned at line 16, the algorithm is not able to conclude— $\mathcal{W}$  may be fully inside the WFW. In this latter case, the algorithm can be executed again with a smaller threshold  $\epsilon$  and with the final list  $\mathcal{L}$  of the previous execution as an input.

### C. Total Orientation WFW

For parallel cable-driven robots having both translational and rotational DOF, the total orientation WFW is defined as the set of positions  $\mathbf{p}$  of the reference point  $P$  such that the



**Input:**  $\mathcal{B}$ ,  $[\psi]$ ,  $[\mathbf{f}]$ ,  $[\boldsymbol{\tau}]$ ,  $\epsilon_{pos}$ ,  $\epsilon_{ori}$   
**Output:**  $\mathcal{L}_{in}$ ,  $\mathcal{L}_{out}$ ,  $\mathcal{L}_{neg}$

1.  $\mathcal{L} \leftarrow \mathcal{B}$  % Initialize list  $\mathcal{L}$  with the search box  $\mathcal{B}$
2. **while**  $\mathcal{L} \neq \{\emptyset\}$  **do**
3.    $[\mathbf{p}] \leftarrow \text{Extract}(\mathcal{L})$
4.    $[\mathbf{x}] = ([\mathbf{p}], [\psi])$
5.    $\text{state} = \text{FullyInside}([\mathbf{x}], [\mathbf{f}], [\boldsymbol{\tau}], \epsilon_{ori})$
6.   **if**  $\text{state} = \text{true}$  **then**
7.      $\mathcal{L}_{in} \leftarrow [\mathbf{p}]$  %  $[\mathbf{p}] \subseteq$  total orientation WFW
8.   **else if**  $\text{state} = \text{false}$  **then**
9.      $\mathcal{L}_{out} \leftarrow [\mathbf{p}]$  %  $[\mathbf{p}] \cap$  (total orientation WFW) =  $\{\emptyset\}$
10.   **else** %  $\text{state} = \text{unknown}$
11.     **if**  $\text{Size}([\mathbf{p}]) > \epsilon_{pos}$  **then**
12.        $\mathcal{L} \leftarrow \text{Bisect}([\mathbf{p}])$
13.     **else** %  $[\mathbf{p}]$  is too small to be bisected
14.        $\mathcal{L}_{neg} \leftarrow [\mathbf{p}]$
15.     **end if**
16.   **end if**
17. **end while**

Fig. 6. A basic algorithm to determine the total orientation WFW.

mobile platform pose  $\mathbf{x} = (\mathbf{p}, \psi)$  belongs to the WFW for any orientation  $\psi$  taken in a given set of orientations  $[\psi]$ —which, in this paper, is a box. The total orientation WFW is of interest because it can be visualized. Indeed, it is a subset of the Cartesian space of the robot (the space of the positions of the reference point  $P$ ).

The pseudocode shown in Fig. 6 is a basic algorithm that computes the total orientation WFW by means of a set of position boxes  $[\mathbf{p}]$ . The output list  $\mathcal{L}_{in}$  contains these latter boxes whereas  $\mathcal{L}_{out}$  contains boxes covering the exterior of the total orientation WFW. The output list  $\mathcal{L}_{neg}$  contains the boxes  $[\mathbf{p}]$  for which the algorithm cannot conclude. The important fact about  $\mathcal{L}_{neg}$  is that it always contains the boundary of the total orientation WFW since the procedures `Feasible()` and `Out()`—which are used in `FullyInside()`—never return true when applied to a box containing a part of the WFW boundary. The procedure `FullyInside()` called at line 5 is a slight modification of the algorithm shown in Fig. 5 in which the bisection (line 13 in Fig. 5) is only applied to the orientation part of the current box  $[\mathbf{x}]$  with associated threshold  $\epsilon_{ori}$ . `FullyInside()` returns **true** if  $[\mathbf{x}]$  is fully included in the WFW, **false** if a part of  $[\mathbf{x}]$  is found to be outside the WFW and **unknown** otherwise.

#### D. On the Effect of the Bisections

In the algorithms presented in Fig. 4 to 6, the repeated bisections progressively reduce the size of the boxes  $[\mathbf{x}]$  (or  $[\mathbf{p}]$ ), stored in the list  $\mathcal{L}$ , for which the procedures `Feasible()` and `Out()` (or `FullyInside()`) cannot conclude. Ultimately, the current box  $[\mathbf{x}]$  would be reduced to a single pose  $\mathbf{x}$  and the interval wrench matrix  $[\mathbf{W}] = [\mathbf{W}]([\mathbf{x}])$  would consequently be reduced to the wrench matrix  $\mathbf{W}$  obtained for  $\mathbf{x}$ . According to the definition of the WFW stated in Section II, for such a unique pose  $\mathbf{x}$  and associated wrench matrix  $\mathbf{W}$ , (17) (resp. (26)) is necessary and sufficient for  $\mathbf{x}$  to belong to the WFW (resp. to be outside the WFW), whereas it is only sufficient in the case of a non-degenerate box  $[\mathbf{x}]$ . This characteristic of (17) and (26) is a key property that makes the branch-and-prune algorithms presented in this section work well.

In order to emphasize this important point, consider for instance the condition (17). For a box  $[\mathbf{x}]$  which is fully included in the WFW, (17) may not be verified because the enclosure  $[\mathbf{W}]$  overestimates the set of wrench matrices (11), *i.e.*, because of the wrapping effect. After the bisection of  $[\mathbf{x}]$  which yields two smaller boxes, say  $[\mathbf{x}]_1$  and  $[\mathbf{x}]_2$ , the overestimations involved in the computations of the interval wrench matrices  $[\mathbf{W}]_1 = [\mathbf{W}]_1([\mathbf{x}]_1)$  and  $[\mathbf{W}]_2 = [\mathbf{W}]_2([\mathbf{x}]_2)$  are somehow lesser and the condition (17) has more chance of being true for both  $[\mathbf{x}]_1$  and  $[\mathbf{x}]_2$  than for their parent box  $[\mathbf{x}]$ .

Finally, note that the strategy used for choosing the bisected variable can play an important role as an appropriate choice may avoid a large number of bisections. In the present work, only basic strategies have been tested such as the round-robin strategy in which variables are bisected alternately. Other strategies may turn out to be useful, *e.g.*, the maximum smear heuristic [48].

## VII. IMPROVEMENT OF EFFICIENCY

The algorithms presented in the previous section are basic. In order to improve their efficiency, many tricks can be employed. For instance, in the algorithm shown in Fig. 5, one can first verify if one of the vertices of the prescribed workspace  $\mathcal{W}$  is outside the WFW. Indeed, in this case, it is useless to explore the whole workspace  $\mathcal{W}$  with a branch-and-prune algorithm. Besides, the efficiency of the proposed methods can also be improved by modifying equation (1) with which we work from the outset. The gain in efficiency is substantial enough to deserve a discussion which is the purpose of the present section.

### A. Wrench Matrix Without Denominators

The computation time of the algorithms presented in Section VI is highly dependent upon the quality of the enclosure of the set (11) by the interval wrench matrix  $[\mathbf{W}]$ . Sharper enclosures lead to a lower number of bisections and thus to a lower computation time. To this end, this section presents modifications of the problem which allows us to work with a matrix  $\mathbf{W}^*$  whose symbolic expression is simpler than that of  $\mathbf{W}$ . Indeed, due to the so-called dependence problem [40]–[42], the quality of the interval evaluation of an expression depends highly on the number of occurrences of each variable for which the expression is evaluated. The fewer occurrences, the better is the interval evaluation. Hence, it is worth working with a matrix  $\mathbf{W}^*$  whose components  $w_{ij}^*$  have symbolic expressions containing less occurrences of the pose variables than those of  $\mathbf{W}$ . A good candidate for  $\mathbf{W}^*$  is the matrix obtained from the wrench matrix  $\mathbf{W}$  by removing the denominators appearing in each of its elements, *i.e.*

$$\mathbf{W}^* = \mathbf{W}\mathbf{D} \quad (29)$$

where  $\mathbf{D} = \text{diag}(\rho_1, \dots, \rho_m)$  is an  $m \times m$  diagonal matrix with the cable lengths  $\rho_i$  on its diagonal. In the case of six-DOF robots, the column vectors of  $\mathbf{W}^*$  are given by

$$\mathbf{w}_j^* = \begin{pmatrix} \mathbf{l}_j \\ \mathbf{b}_j \times \mathbf{l}_j \end{pmatrix}_{6 \times 1}. \quad (30)$$

Compared to (2), (30) involves  $\mathbf{l}_i$  instead of  $\mathbf{d}_i = \mathbf{l}_i/\rho_i$  such that the denominators  $\rho_i$  have been removed. Since  $\rho_i$  is a function of the pose variables  $\mathbf{x}$ , the expressions of the components of  $\mathbf{W}^*$  contain less occurrences of the pose variables than those of  $\mathbf{W}$ . For a given box of poses  $[\mathbf{x}]$ , the quality of the interval evaluation of  $\mathbf{W}^*$ ,  $[\mathbf{W}^*] = [\mathbf{W}^*](\mathbf{x})$ , is thereby much better than that of  $\mathbf{W}$ , *i.e.* the wrapping effect is reduced.

### B. Modification of the Sufficient Conditions

So as to work with  $\mathbf{W}^*$  instead of  $\mathbf{W}$ , the sufficient condition (17) for a box of poses  $[\mathbf{x}]$  to be fully included in the WFW must be modified as follows

$$\forall \mathbf{W}^* \in [\mathbf{W}^*], \forall \mathbf{f} \in [\mathbf{f}], \exists \boldsymbol{\tau}^* \in [\boldsymbol{\tau}^*] \text{ such that } \mathbf{W}^* \boldsymbol{\tau}^* = \mathbf{f} \quad (31)$$

with the interval vector  $[\boldsymbol{\tau}^*]$  defined by

$$[\boldsymbol{\tau}^*] = \{ \boldsymbol{\tau}^* \mid \tau_i^* \in [\tau_{i_{min}}/\underline{\rho}_i, \tau_{i_{max}}/\overline{\rho}_i], \forall 1 \leq i \leq m \} \quad (32)$$

where  $\underline{\rho}_i$  and  $\overline{\rho}_i$  are, respectively, the lower and upper bounds of the interval evaluation  $[\rho_i]$  of  $\rho_i$  for the box  $[\mathbf{x}]$  ( $[\rho_i] = [\rho_i](\mathbf{x}) = [\underline{\rho}_i, \overline{\rho}_i]$ ) whereas  $\tau_{i_{min}}$  and  $\tau_{i_{max}}$  are defined in (3). In (31), the interval matrix  $[\mathbf{W}^*]$  corresponds to the interval evaluation of  $\mathbf{W}^*$  for the box  $[\mathbf{x}]$ . We shall note that the modified set of allowed cable tensions  $[\boldsymbol{\tau}^*]$  is properly defined only if

$$\tau_{i_{min}}/\underline{\rho}_i \leq \tau_{i_{max}}/\overline{\rho}_i \quad (33)$$

Thus, *the problem can be modified only if (33) is verified for all  $i$* . The fact that (31) is a sufficient condition for  $[\mathbf{x}]$  to be fully included in the WFW is proved in the appendix.

Finally, in order to work with  $\mathbf{W}^*$  instead of  $\mathbf{W}$ , the sufficient condition (26) for a box  $[\mathbf{x}]$  to be fully outside the WFW is to be modified as follows

$$\exists \mathbf{f}_y \in \{\mathbf{f}_y\} \mid \forall \mathbf{W}^* \in [\mathbf{W}^*], \forall \boldsymbol{\tau}^* \in [\boldsymbol{\tau}^*], \mathbf{W}^* \boldsymbol{\tau}^* \neq \mathbf{f}_y \quad (34)$$

where  $[\mathbf{W}^*] = [\mathbf{W}^*](\mathbf{x})$  and  $[\boldsymbol{\tau}^*]$  is defined by

$$[\boldsymbol{\tau}^*] = \{ \boldsymbol{\tau}^* \mid \tau_i^* \in [\tau_{i_{min}}/\overline{\rho}_i, \tau_{i_{max}}/\underline{\rho}_i], \forall 1 \leq i \leq m \} \quad (35)$$

with  $[\underline{\rho}_i, \overline{\rho}_i] = [\rho_i](\mathbf{x})$ . The proof of the fact that (34) is a sufficient condition for  $[\mathbf{x}]$  to be fully outside the WFW resembles that of (31) and is therefore left to the reader. Note that the box  $[\boldsymbol{\tau}^*]$  defined in (35) is not the same as the one defined in (32). Note also that, contrary to (32), no condition such as (33) must be fulfilled for (35) to have a meaning (provided that  $\underline{\rho}_i \neq 0$ ). Hence, one can (and should) always use (34) in place of (26).

## VIII. EXAMPLES

Our implementation in C++ uses the interval arithmetic of the BIAS/Profil C++ library, the simplex method of the GNU Linear Programming Kit (GLPK) and the hull consistency procedure of ALIAS [47]. The computation times have been obtained on a DELL XPS laptop (Core 2 Duo CPU T7500, 2.20 GHz).

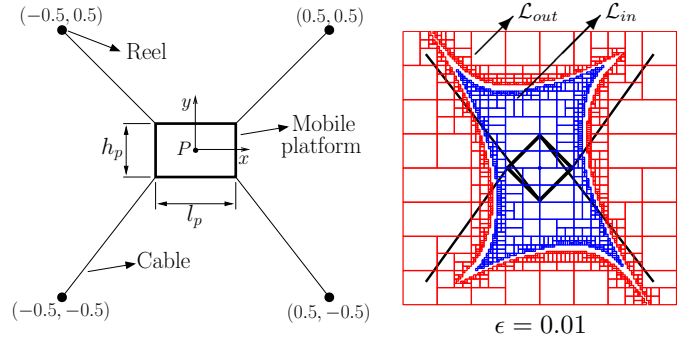


Fig. 7. A 3-DOF planar parallel cable-driven robot (left) and its constant-orientation WFW (right) for orientation  $\phi = \pi/4$  and geometric parameters  $l_p = 0.2$  m,  $h_p = -0.2$  m (crossed cables)—computation time of 2 s (right figure).

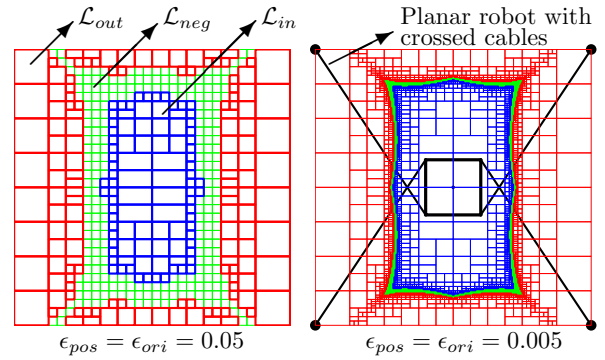


Fig. 8. Total orientation workspace computations for two values of  $\epsilon_{pos}$  and  $\epsilon_{ori}$ —computation time of 2 s (left figure) and 51 s (right figure).

### A. 3-DOF Planar Robots

Let us consider the 3-DOF planar robot driven by four cables shown in Fig. 7 (left) in its reference orientation. The actuated reels fixed to the base are located at the vertices of a square of side length 1 m. The mobile platform is a rectangle of length  $l_p$  and height  $h_p$ . We let  $l_p$  and  $h_p$  be negative in order to consider robots with crossed cables. For instance, the robot obtained for  $l_p = 0.2$  m and  $h_p = -0.2$  m is shown in Fig. 8 together with its total orientation WFW obtained by the algorithm of Section VI-C for the set of orientations  $[-\pi/5, \pi/5]$ . The WFW is defined here by  $\tau_{min}=1$  N and  $\tau_{max}=50$  N for each cable and  $[\mathbf{f}]=[-10, 10], [-10, 10], [-0.5, 0.5]^T$  (N,N,N.m). A constant-orientation WFW obtained with the algorithm of Section VI-A is shown in the right subfigure of Fig. 7. Using the pseudocode of Section VI-B, for  $l_p = 0.2$  m, the box of poses  $[\mathbf{x}]=[-0.2, 0.2], [-0.2, 0.2], [-\pi/5, \pi/5]^T$  (m,m,rad) is found not to be fully included in the WFW in less than 1 s for  $h_p = 0.2$ ,  $h_p = 0.1$  and  $h_p = 0$  (m). For  $h_p = -0.1$  and  $h_p = -0.2$  (m), this box is found to be fully inside the WFW in 2 s.

### B. A 6-DOF Robot

The 6-DOF parallel cable-driven robot considered is shown in Fig. 9 in its reference orientation. Its base frame is a cube of edge length 1 m. Its mobile platform, a regular tetrahedron of

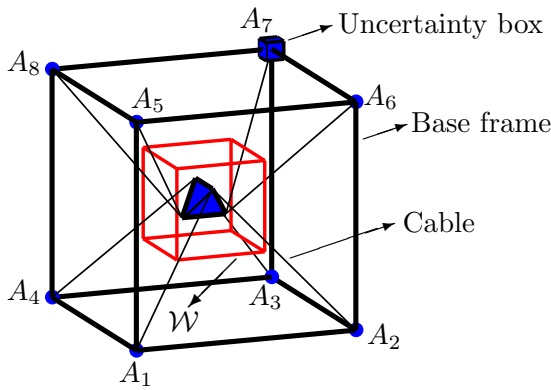


Fig. 9. A six-DOF parallel robot driven by eight cables.

	$l = 0.2$ m $s = 0.2$ m (IN)	$l = 0.1$ m $s = 0.4$ m (IN)	$l = 0.2$ m $s = 0.4$ m (IN)	$l = 0.2$ m $s = 0.6$ m (OUT)
with $\mathbf{W}^*$ (Section VII)	1 s	41 s	47 s	< 1 s
with $\mathbf{W}$	7 s	574 s	663 s	< 1 s

TABLE I

COMPUTATION TIMES FOR  $\epsilon = 0.01$  (THE MENTION “IN” MEANS THAT  $\mathcal{W}$  IS FULLY INCLUDED IN THE WFW WHEREAS “OUT” MEANS THAT IT IS NOT).

edge length  $l$ , is driven by eight cables. Let us also consider the WFW defined by  $\tau_{min}=1$  N,  $\tau_{max}=540$  N for each cable and  $[\mathbf{f}]=[[[-10, 10], [-10, 10], [-10, 10], [-0.5, 0.5], [-0.5, 0.5], [-0.5, 0.5]]^T$  (forces in N, moments in N.m) together with a prescribed workspace  $\mathcal{W}$ . The position components of  $\mathcal{W}$  form a cube of edge length  $s$  whose center coincides with that of the base frame whereas its orientation part is a box of three Euler angles  $\phi$ ,  $\theta$  and  $\psi$  (XYZ convention) such that  $\phi \in [-\pi/12, \pi/12]$ ,  $\theta \in [-\pi/12, \pi/12]$  and  $\psi = 0$  (rad).  $\mathcal{W}$  is thus a five-dimensional workspace.

The computation times of our implementation of the pseudocode of Fig. 5 that determines whether  $\mathcal{W}$  is fully included in the WFW are shown in Table I for various values of  $l$  and  $s$ . Table II shows examples of computation times necessary to test the poses of a discretization of  $\mathcal{W}$  for various values of the number of points used to discretize each axis. The results are consistent as all the poses of the various discretization grids are found to lie in the WFW. Comparing these results with those of Table I shows that, in this example, the interval analysis based method (computation time of 47 s) is more efficient than a discretization with at least 6 points per axis which is a remarkable result since the infinitely many poses of  $\mathcal{W}$  have been verified to be wrench feasible while only finitely many of them are checked by a discretization.

Fig. 10 shows the total orientation WFW obtained by the algorithm of Section VI-C for the set of orientations  $\phi \in [-\pi/12, \pi/12]$ ,  $\theta \in [-\pi/12, \pi/12]$  and  $\psi = 0$  (rad). For clarity, only the boxes of  $\mathcal{L}_{in}$  are shown. The considered search space (box  $\mathcal{B}$ ) is the whole base frame cube. The corresponding computation time is 1567 s. Despite this computation time value, the total volume of the boxes in  $\mathcal{L}_{neg}$ —0.6 m<sup>3</sup> for a search space volume of 1 m<sup>3</sup>—remains quite large. This

	$p_i = 4$ $p = 1024$	$p_i = 6$ $p = 7776$	$p_i = 8$ $p = 32768$	$p_i = 10$ $p = 100\,000$
Computation time	11 s	84 s	352 s	1077 s

TABLE II

EXECUTION TIME FOR VARIOUS DISCRETIZATIONS OF  $\mathcal{W}$  (FOR  $l = 0.2$  AND  $s = 0.4$ ) WHERE  $p_i$  DENOTES THE NUMBER OF POINTS IN THE DISCRETIZATION OF EACH AXIS OF  $\mathcal{W}$  AND  $p$  THE TOTAL NUMBER OF POINTS (POSES) IN THE DISCRETIZATION GRID,  $p = p_i^5$ .

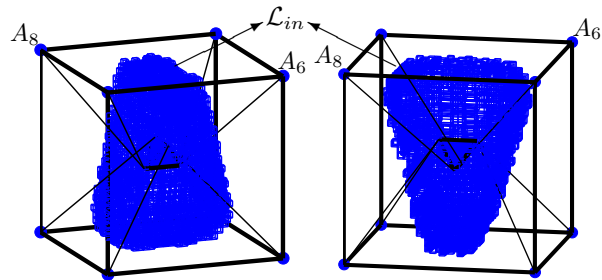


Fig. 10. Total orientation workspace computation for  $\epsilon_{pos} = \epsilon_{ori} = 0.05$  and  $l = 0.2$  m ( $\mathcal{L}_{in}$ ,  $\mathcal{L}_{out}$  and  $\mathcal{L}_{neg}$  total volumes are equal to 0.22 m<sup>3</sup>, 0.18 m<sup>3</sup> and 0.6 m<sup>3</sup>, respectively).

is an example in which the wrapping effect has a negative impact on the efficiency since most of the boxes in  $\mathcal{L}_{neg}$  do not contain any boundary pose of the total orientation WFW (boxes referred to as being of the “second type” in Section VI-A).

### C. Including Uncertainties

Let us consider again the WFW, the prescribed workspace  $\mathcal{W}$  and the 6-DOF parallel cable robot of Section VIII-B (for  $l = 0.2$  m). Let us assume that each  $\mathbf{a}_i$  and each  $\mathbf{b}_i$ , which define the geometry of the robot (see Fig. 2), are not exactly known. Instead, the end point of each of these vectors is only known to lie in a small box. Such an uncertainty box is illustrated in Fig. 9 in the case of  $\mathbf{a}_7$ . In Section III-B, a vector  $\mathbf{u}$  collecting all the geometric design parameters has been defined. The uncertainties on  $\mathbf{a}_i$  and  $\mathbf{b}_i$  yields a (small) box  $[\mathbf{u}]$  of uncertain design parameters.

The procedures of Section VI can easily be modified so as to handle such boxes  $[\mathbf{u}]$ . For instance, in the algorithm of Fig. 5, at line 6, it suffices to compute the interval wrench matrix  $[\mathbf{W}]$  by interval evaluating  $\mathbf{W}$  for the current box  $[\mathbf{x}]$  and for  $[\mathbf{u}]$ . Note that, since  $\forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{u} \in [\mathbf{u}], \mathbf{W}(\mathbf{x}, \mathbf{u}) \in [\mathbf{W}]$ , when Feasible() (resp. Out()) returns **true**, it means that for all  $\mathbf{u}$  in  $[\mathbf{u}]$ , the box of poses  $[\mathbf{x}]$  is fully included in the WFW (resp. fully outside). In other words,  $[\mathbf{x}]$  is guaranteed to be fully included in the WFW (resp. fully outside) for any value of the geometric design parameters in their respective uncertainty boxes.

For example, for uncertainty boxes of edge lengths 0.002 m, 0.004 m and 0.006 m, a modified version of the algorithm of Fig. 5 shows that  $\mathcal{W}$  is fully included in the WFW in 128 s, 297 s and 1095 s, respectively. For larger uncertainty boxes, because bisections in the high-dimensional space of geometric design parameters would be required ( $\mathbf{u}$  has dimension 48 for a

6-DOF robot with 8 cables), the computation time can be very high. Note however that, even for small uncertainty boxes for which the method proposed in this paper is able to conclude, a discretization of each uncertainty box (even very coarse), besides providing no guarantee on the result validity, leads to prohibitive computation times due the dimension of  $\mathbf{u}$ .

## IX. CONCLUSION

An interval analysis based approach to the WFW determination of  $n$ -DOF parallel robots driven by  $n$  or more than  $n$  cables has been proposed. The WFW considered is such that the required wrench set is a box. The main novelty of our approach is to provide two tests. The first one enables to find boxes guaranteed to be fully inside the WFW whereas the second one enables to detect boxes guaranteed to be fully outside the WFW. The paper has then discussed the use of these two tests within standard branch-and-prune algorithms. Moreover, the use of a wrench matrix without denominators has been proposed as a means of reducing the so-called wrapping effect thereby significantly improving computation time.

The distinctive characteristic of our approach is that none of the infinitely many poses of the search space is disregarded. The results are thus always of better quality since full-dimensional sets of poses (here boxes) are returned, in comparison to a discrete finite set of individual poses in the case of a discretization. Let us also point out that testing all the poses of a fine enough discretization grid always leads to a computation time higher than the one of the approach proposed in this paper. Besides, our approach turns out to be particularly suited to test whether a given prescribed workspace is fully included in the WFW, a problem for which it offers competitive computation times. Hence, it is a valuable tool for the dimensional synthesis framework introduced in [49]. Additionally, it has been briefly illustrated how the proposed approach can deal with small uncertainties on the geometric design parameters of a parallel cable robot. A drawback of the proposed approach is that its computation time can hardly be predicted.

The approach to the WFW determination proposed in this paper may also return boxes of poses for which no conclusion could be drawn (at the current algorithm resolution). The set of these boxes has the important property of always containing the part of the WFW boundary included in the search space thereby providing a conservative approximation of this boundary. Note however that, among these boxes, some of them are either fully inside or else fully outside the WFW but not detected as such due to the wrapping effect. In this paper, we have proposed and shown how to work with a wrench matrix without denominators so as to reduce this effect. Further improving efficiency by mitigating the negative consequences of the wrapping effect remains an important research issue and is part of our future works.

## APPENDIX

*Proof of (31):* let  $\mathbf{x}$  be any pose lying in the box  $[\mathbf{x}]$ . According to (31), for each  $\mathbf{f} \in [\mathbf{f}]$  there exists a  $\boldsymbol{\tau}^*$  in  $[\boldsymbol{\tau}^*]$

such that  $\mathbf{W}^* \boldsymbol{\tau}^* = \mathbf{f}$  where  $\mathbf{W}^* = \mathbf{W}^*(\mathbf{x})$ . If  $\mathbf{W} = \mathbf{W}(\mathbf{x})$  is the wrench matrix associated with the pose  $\mathbf{x}$ , according to (29), we have  $\mathbf{W}^* = \mathbf{W}\mathbf{D}$ . Then,  $\mathbf{W}^* \boldsymbol{\tau}^* = \mathbf{f}$  implies that

$$\mathbf{W}\boldsymbol{\tau} = \mathbf{f} \text{ with } \boldsymbol{\tau} := \mathbf{D}\boldsymbol{\tau}^* = (\rho_1\tau_1^*, \dots, \rho_m\tau_m^*)^T \quad (36)$$

where  $\rho_i$  is the length of cable  $i$  for the pose  $\mathbf{x}$  at hand. Now, since  $\boldsymbol{\tau}^*$  is in  $[\boldsymbol{\tau}^*]$ , for each of its component  $\tau_i^*$  we can write

$$\tau_{i_{min}}/\underline{\rho}_i \leq \tau_i^* \leq \tau_{i_{max}}/\overline{\rho}_i. \quad (37)$$

Moreover, by definition of  $\underline{\rho}_i$  and  $\overline{\rho}_i$ , for each  $i$  we have  $0 \leq \underline{\rho}_i \leq \rho_i \leq \overline{\rho}_i$  which, assuming  $\underline{\rho}_i > 0$  (i.e.,  $\underline{\rho}_i \neq 0$ ), implies

$$1 \leq \rho_i/\underline{\rho}_i \text{ and } \rho_i/\overline{\rho}_i \leq 1 \quad (38)$$

and consequently

$$\tau_{i_{min}} \leq (\tau_{i_{min}} \rho_i)/\underline{\rho}_i \text{ and } (\tau_{i_{max}} \rho_i)/\overline{\rho}_i \leq \tau_{i_{max}} \quad (39)$$

since  $\tau_{i_{max}} \geq \tau_{i_{min}} \geq 0$ . Multiplying both inequalities of (37) by  $\rho_i$ , according to (39) and since  $\tau_i = \rho_i\tau_i^*$ , we obtain

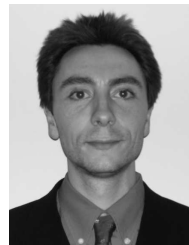
$$\tau_{i_{min}} \leq \tau_i \leq \tau_{i_{max}}. \quad (40)$$

In other words,  $\boldsymbol{\tau}$  is in  $[\boldsymbol{\tau}]$ , the system  $\mathbf{W}\boldsymbol{\tau} = \mathbf{f}$  is feasible and, consequently,  $\mathbf{x}$  belongs to the WFW. Since this reasoning is valid for any pose  $\mathbf{x}$  in  $[\mathbf{x}]$ , (31) is a sufficient condition for  $[\mathbf{x}]$  to be fully included in the WFW.  $\square$

## REFERENCES

- [1] S. Kawamura, H. Kino, and C. Won, "High-speed manipulation by using parallel wire-driven robots," *Robotica*, vol. 18, pp. 13–21, 2000.
- [2] R. Dekker, A. Khajepour, and S. Behzadipour, "Design and testing of an ultra-high-speed cable robot," *Int. J. Robotics and Automation*, vol. 21, no. 1, pp. 25–34, 2006.
- [3] M. Ishii and M. Sato, "A 3D spatial interface device using tensed strings," *Presence*, vol. 3, no. 1, pp. 81–86, 1994.
- [4] R. L. Williams II, M. Xin, and P. Bosscher, "Contour-crafting-cartesian-cable robot system concepts: Workspace and stiffness comparisons," in *Proc. ASME Int. Des. Eng. Tech. Conf. and Comp. Inf. in Eng. Conf.*, no. Paper DETC2008-49478, Brooklyn, NY, Aug. 2008.
- [5] S. Perreault and C. M. Gosselin, "Cable-driven parallel mechanisms: Application to a locomotion interface," in *Proc. ASME Int. Des. Eng. Tech. Conf. and Comp. Inf. in Eng. Conf.*, no. Paper DETC2007-35582, Las Vegas, Nevada, Sep. 2007.
- [6] R. Kurtz and V. Hayward, "Dexterity measure for tendon actuated parallel mechanisms," in *Proc. IEEE Int. Conf. Advanced Robotics*, Pisa, Italy, 1991, pp. 1141–1146.
- [7] S. E. Landsberger and T. B. Sheridan, "A minimal minimal linkage: The tension-compression parallel link manipulator," in *Proc. IMACS/SICE Int. Symp. on Robot., Mechatron., and Manuf. Syst.*, Kobe, Japan, Sep. 1993, pp. 81–88.
- [8] J. Albus, R. Bostelman, and N. Dagalakis, "The NIST ROBOCRANE," *J. Robotic Syst.*, vol. 10, no. 5, pp. 709–724, 1993.
- [9] A. Ming and T. Higuchi, "Study on multiple degree-of-freedom positioning mechanism using wires (part 1) - concept, design and control," *Int. J. Japan Society for Precision Eng.*, vol. 28, no. 2, pp. 131–138, 1994.
- [10] R. G. Roberts, T. Graham, and T. Lippitt, "On the inverse kinematics, statics, and fault tolerance of cable-suspended robots," *J. of Robot. Syst.*, vol. 15, no. 10, pp. 581–597, 1998.
- [11] Y. Takeda and H. Funabashi, "A transmission index for in-parallel wire-driven mechanisms," *JSME international journal, Series C*, vol. 41, no. 1, pp. 180–187, 2001.
- [12] R. Verhoeven, "Analysis of the workspace of tendon-based Stewart-platforms," Ph.D. dissertation, Univ. Duisburg-Essen, Germany, 2004.
- [13] R. L. Williams II and P. Gallina, "Planar cable-direct-driven robots: Design for wrench exertion," *J. of Int. and Robot. Syst.*, vol. 35, no. 2, pp. 203–219, 2002.

- [14] K. Maeda, S. Tadokoro, T. Takamori, M. Hiller, and R. Verhoeven, "On design of a redundant wire-driven parallel robot WARP manipulator," in *Proc. IEEE Int. Conf. Robotics and Automation*, Detroit, Michigan, May 1999, pp. 895–900.
- [15] Y. Wischnitzer, N. Shvalb, and M. Shoham, "Wire-driven parallel robot: Permitting collisions between wires," *The Int. J. Rob. Research*, vol. 27, no. 9, pp. 1007–1026, Sep. 2008.
- [16] J.-P. Merlet and D. Daney, "Legs interference checking of parallel robots over a given workspace or trajectory," in *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, Florida, may 2006, pp. 757–762.
- [17] I. Ebert-Uphoff and P. A. Voglewede, "On the connections between cable-driven robots, parallel manipulators and grasping," in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, 2004, pp. 4521–4526.
- [18] P. Bosscher, A. T. Riechel, and I. Ebert-Uphoff, "Wrench-feasible workspace generation for cable-driven robots," *IEEE Trans. on Robotics*, vol. 22, no. 5, pp. 890–902, Oct. 2006.
- [19] A. Fattah and S. K. Agrawal, "On the design of cable-suspended planar parallel robots," *ASME J. Mech. Des.*, vol. 127, no. 5, pp. 1021–1028, 2005.
- [20] G. Barrette and C. M. Gosselin, "Determination of the dynamic workspace of cable-driven planar parallel mechanisms," *ASME J. Mech. Des.*, vol. 127, no. 2, pp. 242–248, 2005.
- [21] M. Gouttefarde and C. M. Gosselin, "Analysis of the wrench-closure workspace of planar parallel cable-driven mechanisms," *IEEE Trans. on Robotics*, vol. 22, no. 3, pp. 434–445, Jun. 2006.
- [22] E. Stump and V. Kumar, "Workspaces of cable-actuated parallel manipulators," *ASME J. Mech. Des.*, vol. 128, no. 1, pp. 159–167, 2006.
- [23] P. Gallina and G. Rosati, "Manipulability of a planar wire driven haptic device," *Mech. and Mach. Theory*, vol. 37, no. 2, pp. 215–228, Feb. 2002.
- [24] P. Voglewede and I. Ebert-Uphoff, "Application of the antipodal grasp Theorem to cable-driven robots," *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 713–718, Aug. 2005.
- [25] C. B. Pham, S. H. Yeo, G. Yang, M. S. Kurbanhusen, and I.-M. Chen, "Force-closure workspace analysis of cable-driven parallel mechanisms," *Mech. and Mach. Theory*, vol. 41, no. 1, pp. 53–69, Jan. 2006.
- [26] C. Ferraresi, M. Paoloni, and F. Pescarmona, "A new methodology for the determination of the workspace of six-DOF redundant parallel structures actuated by nine wires," *Robotica*, vol. 25, no. 1, pp. 113–120, Jan. 2007.
- [27] X. Diao and O. Ma, "A method of verifying force-closure condition for general cable manipulators with seven cables," *Mech. and Mach. Theory*, vol. 42, no. 12, pp. 1563–1576, Dec. 2007.
- [28] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
- [29] S. Behzadipour and A. Khajepour, "Design of reduced DOF parallel cable-based robots," *Mech. and Mach. Theory*, vol. 39, pp. 1051–1065, 2004.
- [30] M. Hassan and A. Khajepour, "Optimization of actuator forces in cable-based parallel manipulators using convex analysis," *IEEE Trans. on Robotics*, vol. 24, no. 3, pp. 736–740, Jun. 2008.
- [31] S. Krut, O. Company, and F. Pierrot, "Force performance indexes for parallel mechanisms with actuation redundancy, especially for parallel wire-driven manipulators," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 3936–3941.
- [32] C. B. Pham, S. H. Yeo, and G. Yang, "Tension analysis of cable-driven parallel mechanisms," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Edmonton, Alberta, Canada, 2005, pp. 257–262.
- [33] S. Bouchard, C. M. Gosselin, and B. Moore, "On the ability of a cable-driven robot to generate a prescribed set of wrenches," in *Proc. ASME Int. Des. Eng. Tech. Conf. and Comp. Inf. in Eng. Conf.*, no. Paper DETC2008-49518, Brooklyn, NY, Aug. 2008.
- [34] M. Fiedler, J. Nedoma, J. Ramik, J. Rohn, and K. Zimmermann, *Linear Optimization Problems with Inexact Data*. Springer, 2006.
- [35] M. Gouttefarde, J.-P. Merlet, and D. Daney, "Wrench-feasible workspace of parallel cable-driven mechanisms," in *IEEE Int. Conf. Robotics and Automation*, Rome, Italy, 2007, pp. 1492–1497.
- [36] T. Bruckmann, L. Mikelsons, M. Hiller, and D. Schramm, "Continuous workspace analysis, synthesis and optimization of wire robots," in *Proc. ASME Int. Des. Eng. Tech. Conf. and Comp. Inf. in Eng. Conf.*, no. Paper DETC2008-49532, Brooklyn, NY, Aug. 2008.
- [37] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitz, "Design, analysis and realization of tendon-based parallel manipulators," *Mech. and Mach. Theory*, vol. 40, no. 4, pp. 429–445, Apr. 2005.
- [38] J.-P. Merlet, *Parallel Robots*, 2nd ed. Springer, 2006.
- [39] R. E. Moore, *Methods and Applications of Interval Analysis*. SIAM Studies in Applied Mathematics, 1979.
- [40] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer Verlag, 2001.
- [41] A. Neumaier, *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.
- [42] E. Hansen and G. W. Walster, *Global Optimization Using Interval Analysis*, 2nd ed. Marcel Dekker, 2004.
- [43] V. Chvátal, *Linear Programming*. New York: W. H. Freeman, 1983.
- [44] A. Neumaier and O. Shcherbina, "Safe bounds in linear and mixed-integer linear programming," *Math. Programming A*, vol. 99, pp. 283–296, 2004.
- [45] O. Lhomme, "Consistency techniques for numeric CSPs," in *Proc. Inter. Joint Conf. on Artificial Intelligence (IJCAI)*, Chambéry, France, 1993, pp. pp. 232–238.
- [46] H. Collavizza, F. Delobel, and M. Rueher, "Comparing partial consistencies," *Reliable Computing*, vol. 5, no. 3, pp. 213–228, 1999.
- [47] J.-P. Merlet, *ALIAS, An Algorithms Library of Interval Analysis for Equation Systems*, [Online]. Available: <http://www-sop.inria.fr/coprin/logiciels/ALIAS/>.
- [48] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*. Dordrecht, Netherlands: Kluwer Academic Publishers, 1996.
- [49] F. Hao and J.-P. Merlet, "Multi-criteria optimal design of parallel manipulators based on interval analysis," *Mech. and Mach. Theory*, vol. 40, no. 2, pp. 157–171, Feb. 2005.



**Marc Gouttefarde** (S'03-M'06) received the B.Eng. degree in mechatronics from the Institut National des Sciences Appliquées (INSA), Strasbourg, France, in 2001, and the M.Sc. and Ph.D. degrees in mechanical engineering from Laval University, Québec, Canada, in 2002 and 2005, respectively. From 2005 to 2007, he was a Postdoctoral Fellow at INRIA with the COPRIN project-team, Sophia-Antipolis, France. He is currently a CNRS Tenured Research Scientist with the LIRMM, Montpellier, France. His main research interests include the design and control of parallel manipulators and parallel cable-driven robots.



**David Daney** received his PhD degree in robotics from the University of Nice Sophia Antipolis, France in 2000. He spent two years as a postdoctoral associate in C.M.W., France, to study a PKM milling machine (2000) and in the LORIA Lab, Nancy, France, on computer arithmetics (2001). In 2002, he has been invited by Mc Gill University, Rutgers University and Laval University. Since 2003 he is an INRIA research scientist, and currently in the COPRIN project-team. His research interests are in the area of robotics, analysis, design, and calibration of parallel robots. More recently, he studies interval analysis and its applications to robotics, in particular, to wire-driven robot performance analysis. Since 2010, he is the leader of an "INRIA large scale initiative action" on the autonomy for elderly people in long-term care.



**Jean-Pierre Merlet** (M'01) is senior researcher at INRIA Sophia Antipolis - Mediterranée, where he leads the COPRIN project-team. His research interest are interval analysis (for the management of uncertainties in the design and control of robots), parallel robots and assistive robotics. J.-P. Merlet was an Associate Editor of the IEEE Transactions on Robotics and is currently Associate Editor of Mechanism and Machine Theory and of the ASME Journal of Mechanisms and Robotics. He was General Chair of the 2007 IFToMM World Congress and of the 2008 IEEE/RSJ IROS Conference.