Technical Report

IRI-TR-15-04

Computing Wrench-feasible Paths for Cable-driven Hexapods

Oriol Bohigas
Montserrat Manubens
Lluís Ros

April, 2015

Institut de Robòtica i Informàtica Industrial

Abstract

Motion paths of cable-driven hexapods must carefully be planned to ensure that the lengths and tensions of all cables remain within acceptable limits, for a given wrench applied to the platform. The cables cannot go slack –to keep the control of the robot– nor excessively tight –to prevent cable breakage– even in the presence of bounded perturbations of the wrench. This paper proposes a path planning method that accommodates such constraints simultaneously. Given two configurations of the robot, the method attempts to connect them through a path that, at any point, allows the cables to counteract any wrench lying in a predefined uncertainty region. The feasible C-space is placed in correspondence with a smooth manifold, which facilitates the definition of a continuation strategy to search this space systematically from one configuration, until the second configuration is found, or path non-existence is proved at the resolution of the search. The force Jacobian is full rank everywhere on the C-space, which implies that the computed paths will naturally avoid crossing the forward singularity locus of the robot. The adjustment of tension limits, moreover, allows to maintain a meaningful clearance relative to such locus. The approach is applicable to compute paths subject to geometric constraints on the platform pose, or to synthesize free-flying motions in the full six-dimensional C-space. Experiments are included that illustrate the performance of the method in a real prototype.

Institut de Robòtica i Informàtica Industrial (IRI)
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona, Spain
Tel (fax): +34 93 401 5750 (5751)
http://www.iri.upc.edu

Corresponding author:
L. Ros
tel: +34 93 401 0775
ros@iri.upc.edu
http://www.iri.upc.edu/staff/ros

1 Introduction

In recent years, cable-driven parallel robots have been increasingly studied and applied to more and more relevant tasks, such as manipulation of heavy loads [12, 34], high-precision positioning [31], monitoring of aquatic environments [8, 23], automated construction of civil structures [10], rescue systems [33], or motion simulators [44]. Among them, hexapodal ones stand out for their simplicity and extensive use, especially after the long-term effort on the NIST Robocrane and its derived applications [1, 11, 15, 31, 32, 41]. They involve the minimum number of cables and motors to fully govern a load in 6D under gravity, resulting in simple robotic cranes for precise manipulation that can even be made mobile by attaching vehicles to the feet (Fig. 1). These advantages, together with the fact that they can easily achieve larger workspaces than their counterparts with rigid-limb legs, make cable-driven hexapods energy-efficient and appropriate to maneuver heavy loads. However, additional constraints apply: their cables can pull but are unable to push the platform, which obliges to keep the cable tensions positive during normal operation.

The C-space of hexapod is limited by a number of hypersurfaces corresponding to configurations where the tension of some cable is either zero, for which the cable goes slack and control of one degree of freedom is lost, or goes to infinity, which indicates that the mechanism is in a singular configuration and the cable can break [45]. In practice, it is important to prevent both extreme situations and ensure that the cables work within a range of admissible tensions, for a given platform wrench subject to bounded perturbations in all directions. After [19] and [9], the configurations fulfilling this condition are said to be *wrench-feasible*.

Several methods have been proposed for the determination of wrench-feasible workspaces of cable-driven parallel robots [9, 22, 37, 43, 45, 46], but the problem of planning paths between given configurations has received little attention comparatively. Algorithms indeed exist that try to avoid the singular configurations where the leg forces tend to infinity, but they are mainly tailored to parallel robots with rigid UPS legs [3, 16, 17, 42], and their application to cable-driven hexapods is not straightforward because they do not account for the positivity constraint on the leg tensions. Moreover, these algorithms measure the clearance of the path relative to the singularity locus using the determinant or the condition number of the Jacobian matrix, which, as noted in [47], lack physical significance. While some path planning approaches apply to cable-driven hexapods [20, 27, 29], the path they compute is evaluated for feasibility at discrete points only, so a method that guarantees the fulfillment of all the constraints along the whole path is still lacking.

The planning method presented in this paper is aimed at covering such gap. It was preliminarly introduced in [5] and it is now presented with thorough detail and illustrative experiments. The method relies on defining a system of equations whose solution manifold corresponds to the six-dimensional wrench-feasible C-space of the hexapod, so that maneuvering through such manifold guarantees singularity avoidance at all times, while maintaining cable tensions and lengths within their allowable bounds (Sections 2 and 3). This manifold, as well as any of its subsets defined by motion constraints arising in many applications, are shown to be smooth everywhere, which is key to define a continuation method able to explore the C-space systematically from one configuration, until a goal configuration is found, or path non-existence is proved at the resolution of the search (Section 4). The method has been implemented and validated in several test cases and with experiments in a real prototype (Section 5), and its main strengths and points for future attention have been identified (Section 6). The method, in conclusion, eliminates the need of current path synthesis approaches based on manual guideance and offline verification.

2 Preliminaries

A cable-driven hexapod consists of a moving platform suspended from a fixed base by means of six cables winding around indepen-

Figure 1: Applications of cable-driven hexapods. From left to right, and top to bottom: Beam manipulation at the NIST Intelligent and Automated Construction Testbed, the ATHLETE Low-gravity Testbed at the Jet Propulsion Lab, fuselage depainting at the NIST Aircraft Maintenance Platform, and a mobile hexapod for oil-well fire fighting.

dent winches (Fig. 2). By actuating the winch drives, the cable lengths d_i can be varied within prescribed limits $(\underline{d_i}, \overline{d_i})$, with $\underline{d_i} > 0$, which allows a full control of the six degrees of freedom of the platform within a given workspace.

The C-space of a cable-driven hexapod can be implicitly defined as follows. Consider fixed and moving reference frames $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively attached to the base and platform links, centered in O and P (Fig. 2). Let $\boldsymbol{p}$ and $\boldsymbol{a}_i$ be the position vectors of P and A_i relative to $\mathcal{F}_1$, and $\boldsymbol{b}_i$ be the position vector of B_i relative to $\mathcal{F}_2$. We can represent any platform configuration by the pair $\boldsymbol{q} = (\boldsymbol{p}, \boldsymbol{R}) \in SE(3) = \mathbb{R}^3 \times SO(3)$, subject to the constraints

$$\boldsymbol{d}_i = \boldsymbol{p} + \boldsymbol{R}\,\boldsymbol{b}_i - \boldsymbol{a}_i, \tag{1}$$

$$d_i^2 = \boldsymbol{d}_i^{\mathsf{T}} \boldsymbol{d}_i, \tag{2}$$

$$\underline{d_i} < d_i < \overline{d_i}, \tag{3}$$

for $i = 1, \ldots, 6$, where $\boldsymbol{R}$ is the 3×3 rotation matrix that provides the orientation of $\mathcal{F}_2$ relative to $\mathcal{F}_1$. While Eqs. (1) and (2) make the cable lengths d_i explicit in terms of $\boldsymbol{p}$ and $\boldsymbol{R}$, the inequalities in (3) constrain such lengths to lie in $(\underline{d_i}, \overline{d_i})$.

The $\boldsymbol{R}$ matrix in Eq. (1) is assumed to be expressed as a function of $\boldsymbol{\tau}$, a tuple of any three angles parameterizing $SO(3)$, such as Euler angles under any convention, or tilt-and-torsion

angles [7]. This allows for an easy formulation of planning problems in constant-angle slices of $SE(3)$, which are useful in parallel kinematic machines [7], and avoids the treatment of additional constraints needed in non-minimal representations of the rotation group. Although we then introduce representation singularities relative to the angles of choice [13, page 31], this will not be problematic because the smoothness properties required to solve our planning problem will remain unaltered.

In practice, any configuration must also be *wrench-feasible*, i.e., it must allow the platform to equilibrate any resultant wrench $\hat{\boldsymbol{w}}$ acting on it, subject to lie inside a prescribed, bounded region $\mathcal{K} \subset \mathbb{R}^6$. The coordinates of $\hat{\boldsymbol{w}}$ are assumed to be given in the usual screw-theoretic form [18], and the significance of $\mathcal{K}$ depends on the particular context of application. In payload transportation, for instance, $\mathcal{K}$ may be given by the gravitational wrench acting on the platform and slight perturbations introduced by inertia forces or external agents like the wind. In contact situations, $\mathcal{K}$ might further depend on the contact wrench against the environment, which is in general subject to six-dimensional uncertainty. Specifically, the wrench-feasibility requirement on a given $\boldsymbol{q}$ implies that for each $\hat{\boldsymbol{w}} \in \mathcal{K}$ there must be a vector of admissible cable tensions

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

satisfying

$$\boldsymbol{J}\,\boldsymbol{f} = \hat{\boldsymbol{w}},$$

where $(\underline{f_i}, \overline{f_i})$ is the range of positive tensions that can be resisted by the ith cable, and $\boldsymbol{J}$ is the 6×6 screw Jacobian of the robot. $\boldsymbol{J}$ is a function of $\boldsymbol{q}$ and takes the form

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_6 \\ \boldsymbol{a}_1 \times \boldsymbol{u}_1 & \cdots & \boldsymbol{a}_6 \times \boldsymbol{u}_6 \end{bmatrix},$$

in frame $\mathcal{F}_1$, where $\boldsymbol{u}_i = \boldsymbol{d}_i / d_i$ [6].

For ease of manipulation, $\mathcal{K}$ will be assumed to be a six-dimensional ellipsoid centered in $\hat{\boldsymbol{w}}_0$, defined implicitly by the inequality

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

where $\boldsymbol{E}$ is a 6×6 positive-definite symmetric matrix. This ellipsoid can be constructed by propagating known bounds on other variables related to $\hat{\boldsymbol{w}}$, using the tools of an ellipsoidal calculus for example [39]. Appendix A shows how to obtain $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ in typical situations, and reveals that both $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ are a function of $\boldsymbol{q}$ in general.

Let us now define the C-space of the manipulator, $\mathcal{C}$, as the set of wrench-feasible configurations $\boldsymbol{q} \in SE(3)$ that satisfy Eqs. (1)–(3) for $i = 1, \ldots, 6$. The planning problem we confront, thus, boils down to computing a *path* joining two given configurations of $\mathcal{C}$, $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$; i.e., a continuous map

$$\boldsymbol{\mu} : [0, 1] \longrightarrow \mathcal{C}$$

such that $\boldsymbol{\mu}(0) = \boldsymbol{q}_s$ and $\boldsymbol{\mu}(1) = \boldsymbol{q}_g$. To tackle this problem, we next define a smooth manifold suitable to navigate $\mathcal{C}$ by numerical continuation [26].

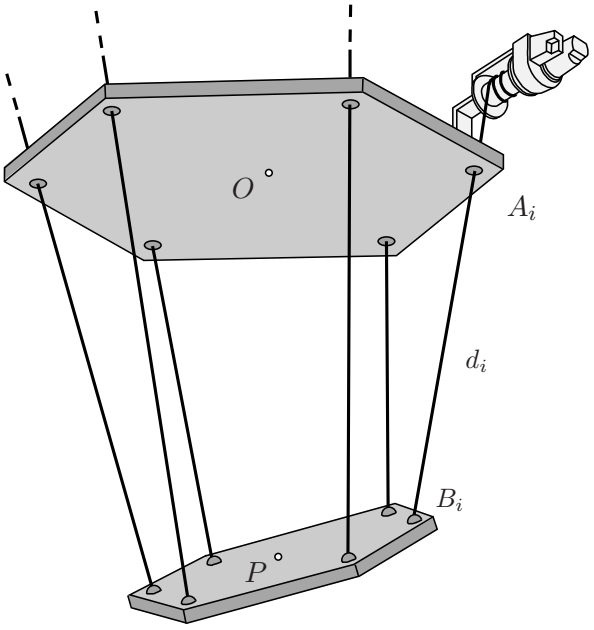


Figure 2: A cable-driven hexapod consists of a moving platform connected to a fixed base by means of six cables. The platform is maintained in a stable position due to the action of gravity.

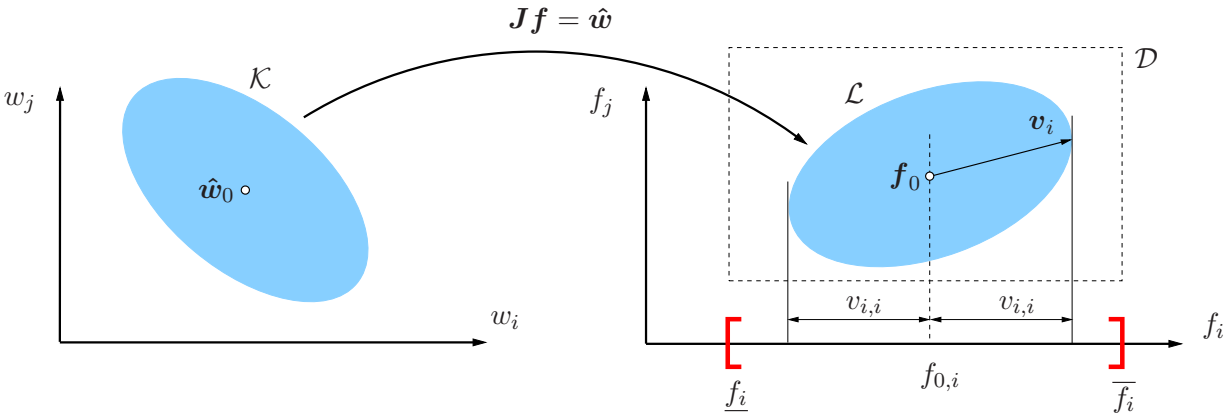


Figure 3: The mapping $\boldsymbol{J}\boldsymbol{f} = \hat{\boldsymbol{w}}$ is used to transform the wrench ellipsoid $\mathcal{K}$ into the cable tension ellipsoid $\mathcal{F}$. The vector $\boldsymbol{v}_i$ provides the maximum and minimum values of f_i within $\mathcal{F}$.

3 The Navigation Manifold

3.1 A Characterization of $\mathcal{C}$

For a given configuration $\boldsymbol{q}$ and a wrench $\hat{\boldsymbol{w}}_0$ applied to the platform, let $\boldsymbol{f}_0$ be the vector of cable tensions corresponding to $\hat{\boldsymbol{w}}_0 \in \mathcal{K}$, which satisfies

$$\boldsymbol{J}\,\boldsymbol{f}_0 = \hat{\boldsymbol{w}}_0. \tag{4}$$

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

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

where $\boldsymbol{B} = \boldsymbol{J}^{\mathsf{T}}\boldsymbol{E}\;\boldsymbol{J}$. This ellipsoid may be bounded in all directions or unbounded in some, depending on whether $det(\boldsymbol{J}) \neq 0$ or not. However, Appendix B shows that $\boldsymbol{J}$ is non-singular for all $\boldsymbol{q} \in \mathcal{C}$, so that $\mathcal{L}$ will always be a bounded ellipsoid in our case (Fig. 3). It is worth to see here that, since $\boldsymbol{J}$ is full rank for all $\boldsymbol{q} \in \mathcal{C}$, the navigation of $\mathcal{C}$ implicitly avoids the singular configurations of the platform. Thus, the control issues related to such configurations (due to output velocity indetermination and platform shakiness [6]) will not be encountered during the execution of the obtained path.

Now, for $\boldsymbol{q}$ to be wrench-feasible, we must have $\mathcal{L} \subseteq \mathcal{D}$, which can be checked as follows. For each $i = 1, \ldots, 6$, let $\boldsymbol{v}_i \in \mathbb{R}^6$ be a vector satisfying

$$\boldsymbol{v}_i^{\mathsf{T}}\boldsymbol{B}\,\boldsymbol{v}_i = 1, \tag{5}$$

$$\boldsymbol{B}^i\boldsymbol{v}_i = \boldsymbol{0}, \tag{6}$$

$$v_{i,i} > 0, \tag{7}$$

where $\boldsymbol{B}^i$ stands for the matrix $\boldsymbol{B}$ with its ith row removed, and $v_{i,i}$ is the i-th component of $\boldsymbol{v}_i$. Observe that if $\boldsymbol{J}$ is non-singular, then both $\boldsymbol{B}$ and $\boldsymbol{B}^i$ are full row rank, and there is exactly one vector $\boldsymbol{v}_i$ satisfying Eqs. (5)-(7). Using Lagrange multipliers, we can see that $\boldsymbol{f}_0 - \boldsymbol{v}_i$ and $\boldsymbol{f}_0 + \boldsymbol{v}_i$ identify the points of $\mathcal{L}$ attaining the smallest and largest value along the ith coordinate (Fig. 3). Hence, when $det(\boldsymbol{J}) \neq 0$, $\mathcal{L}$ is contained in $\mathcal{D}$ whenever

$$f_{0,i} - v_{i,i} > \underline{f_i}, \tag{8}$$

$$f_{0,i} + v_{i,i} < \overline{f_i}, \tag{9}$$

for $i = 1, \ldots, 6$. As a result, $\mathcal{C}$ can be characterized as the set of points $\boldsymbol{q} \in SE(3)$ satisfying Eqs. (1)-(9) for some value of the variables $\boldsymbol{d}_i$, d_i, $\boldsymbol{f}_0$ and $\boldsymbol{v}_i$.

3.2 Conversion into Equality Form

Continuation methods are, by design, aimed at tracing solution sets of systems of equations, not inequalities [26]. To define a continuation-based path planning strategy, we thus need to convert Eqs. (3), and (7)-(9) into equality form.

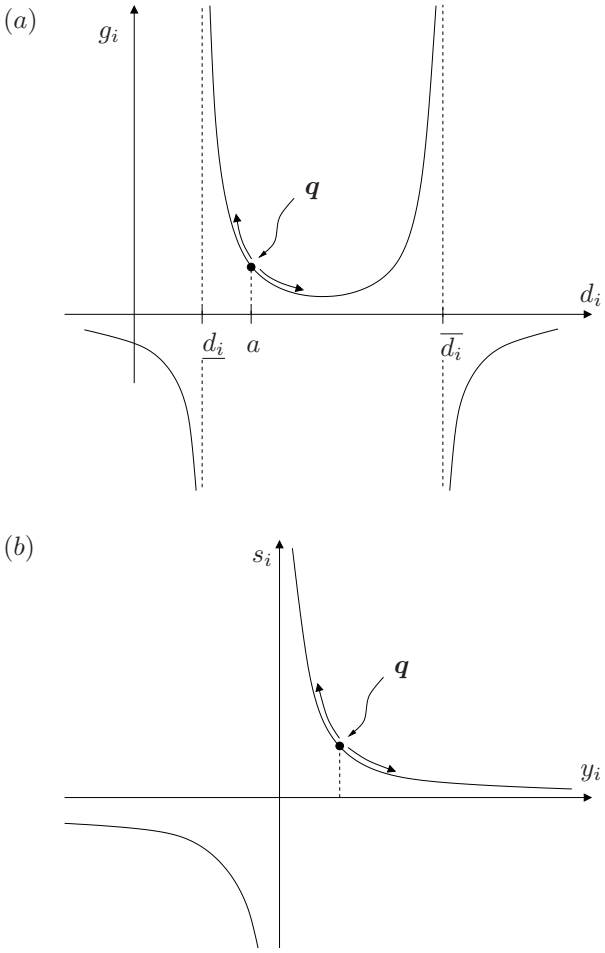


Figure 4: (a) The graph of Eq. (10) shows that if we explore the solution set of Eq. (10) by continuation from some $\boldsymbol{q} \in \mathcal{C}$ corresponding to $d_i = a$, the constraints $\underline{d_i} < d_i < \overline{d_i}$ will always be satisfied. (b) The graph of $y_i s_i = 1$, where $y_i = f_{0,i} - v_{i,i} - \underline{f_i}$, shows that the same applies to Eq. (11).

To this end, note from Fig. 4(a) that we can replace Eq. (3) by

$$(d_i - \underline{d_i}) \cdot (\overline{d_i} - d_i) \cdot g_i = 1, \qquad (10)$$
$$g_i > 0,$$

where g_i is a newly-defined auxiliary variable. In apparence, we have not skipped the use of inequalities with this change, but from the graph of Fig. 4(a) we see that if a configuration $\boldsymbol{q}$ corresponds to a value $d_i = a \in (\underline{d_i}, \overline{d_i})$, then any other configuration found from $\boldsymbol{q}$ by continuation subject to Eq. (10) will always satisfy $\underline{d_i} < d_i < \overline{d_i}$. In other words, the constraint $g_i > 0$ can be neglected under such a continuation scheme.

Similarly, Eqs. (8) and (9) can be replaced by

$$(f_{0,i} - v_{i,i} - \underline{f_i}) \cdot s_i = 1, \qquad (11)$$
$$(\overline{f_i} - f_{0,i} - v_{i,i}) \cdot t_i = 1, \qquad (12)$$
$$s_i > 0, \;\; t_i > 0,$$

where s_i and t_i play a role analogous to that of g_i in Eq. (10). From the graph in Fig. 4(b), for example, it is clear that the quantity $y_i = f_{0,i} - v_{i,i} - \underline{f_i}$ will remain positive, and hence $f_{0,i} - v_{i,i} > \underline{f_i}$, when marching continuously from a given $\boldsymbol{q}$ with $y_i > 0$. The same argument applies to Eq. (12), so we can replace Eqs. (8) and (9) by Eqs. (11) and (12), neglecting the constraints $s_i > 0$ and $t_i > 0$ during the continuation scheme.

Finally, Eq. (7) can be directly neglected, because $\boldsymbol{v}_{i,i} \neq 0$ for all i on any vector satisfying Eqs. (5) and (6). Certainly, observe that $\boldsymbol{B}\boldsymbol{v}_i$ is all zeros except in its i-th component due to Eq. (6). If it were $v_{i,i} = 0$ for some i, this would imply $\boldsymbol{v}_i^{\mathsf{T}} \boldsymbol{B} \boldsymbol{v}_i = 0$, contradicting Eq. (5). Therefore, if our continuation method starts from a value of $\boldsymbol{v}_i$ with $v_{i,i} > 0$, and it is compliant with Eqs. (5) and (6), Eq. (7) will be naturally fulfilled.

3.3 The Manifold and its Properties

The system formed by Eqs. (1)–(2), (4)–(6), and (10)–(12) can be compactly written as

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}, \qquad (13)$$

where $\boldsymbol{x}$ refers to a tuple encompassing all of its variables: $\boldsymbol{p}, \boldsymbol{\tau}, \boldsymbol{d}_i, d_i, g_i, \boldsymbol{f}_0, \boldsymbol{v}_i, s_i$, and t_i, for $i = 1, \ldots, 6$. The solution set of this system,

$$\mathcal{M} = \{\boldsymbol{x} : \boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}\},$$

will be called the *navigation* manifold hereafter, because it has the necessary properties to connect $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ by numerical continuation.

To see this point, consider the subset $\mathcal{M}^+ \subset \mathcal{M}$ formed by the points $\boldsymbol{x}$ for which $g_i > 0$, $s_i > 0$, $t_i > 0$, and $v_{i,i} > 0$. Clearly, $\mathcal{C}$ and $\mathcal{M}^+$

are in correspondence. A point $\boldsymbol{q}$ belongs to $\mathcal{C}$ if, and only if, it has a corresponding point $\boldsymbol{x} \in \mathcal{M}^+$. This implies that any continuous path in $\mathcal{C}$ will also be represented by a continuous path in $\mathcal{M}^+$, and *vice versa*. Thus, the original problem of computing a path of $\mathcal{C}$ connecting $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ can be reduced to that of computing a path in $\mathcal{M}^+$ connecting points $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ of $\mathcal{M}^+$ corresponding to $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$. However, since g_i, s_i, t_i, and $v_{i,i}$ never vanish on $\mathcal{M}$ (Section 3.2), $\mathcal{M}^+$ and its complement $\mathcal{M} \setminus \mathcal{M}^+$ are disconnected, and if we try to connect $\boldsymbol{x}_s$ to $\boldsymbol{x}_g$ by continuation on $\mathcal{M}$, we will be moving through $\mathcal{M}^+$ actually. Therefore, for the purpose of this paper, we will only need $\mathcal{M}$ and Eq. (13) hereafter.

$\mathcal{M}$ is six-dimensional, but Appendix B further proves that it is smooth everywhere, so that every point $\boldsymbol{x}$ has a well-defined tangent space $T_{\boldsymbol{x}}\mathcal{M}$. This greatly simplifies the definition of a continuation method to connect $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$, because no bifurcations, sharpnesses, or dimension changes will be found along the way, avoiding the need of elaborate branch-switching procedures [25].

It is worth adding that in many applications (such as in painting, polishing, or cleaning of ship hulls, wings, or building façades) the platform is further confined to move within a lower-dimensional subset of $\mathcal{C}$ defined by geometric or contact constraints on its pose. As exemplified in Section 5.2, our approach can naturally cope with such constraints, either by including them in Eq. (13) in the parametric form

$$\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{\Omega}(\boldsymbol{\lambda}), \tag{14}$$

where $\boldsymbol{\Omega}$ is a smooth function of any set of parameters $\boldsymbol{\lambda}$, or in the implicit form

$$\boldsymbol{C}(\boldsymbol{p}, \boldsymbol{\tau}) = \boldsymbol{0}, \tag{15}$$

where $\boldsymbol{C}(\boldsymbol{p}, \boldsymbol{\tau})$ is smooth and with a full-rank Jacobian $\boldsymbol{C}_{\boldsymbol{p},\boldsymbol{\tau}}$. Appendix B shows that, again, the resulting system of equations is suitable to the following strategy.

4 A Continuation Strategy

To determine a path connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ we can gradually construct an atlas of the navigation manifold $\mathcal{M}$. An atlas is a collection of smooth maps, called charts, jointly mapping a relatively large subset of $\mathcal{M}$ [30]. In our case, we shall generate the charts starting from $\boldsymbol{x}_s$, trying to map the whole connected component that is reachable from such point, until $\boldsymbol{x}_g$ is eventually found, or path non-existence is proved by exhaustion of the search. Since all variables in $\boldsymbol{x}$ have bounded feasibility intervals, the construction of the atlas will be restricted to a finite domain defined by the Cartesian product of all such intervals. The atlas will be computed using the higher-dimensional continuation method by Henderson [24], with the extensions proposed in [4] to heuristically guide the search towards $\boldsymbol{x}_g$. To make the paper as self-contained as possible, we next recall the main points of the method and refer the reader to [4, 24] for further details.

4.1 Defining a Chart

Let n be the dimension of $\mathcal{M}$, and m be the number of variables in $\boldsymbol{x}$. For a given point $\boldsymbol{x}_i$ of $\mathcal{M}$, set initially to $\boldsymbol{x}_s$, a chart C_i is a local map ψ_i from a polytopic domain $\mathcal{P}_i \in \mathbb{R}^n$ to an open neighbourhood of $\mathcal{M}$ around $\boldsymbol{x}_i$, satisfying $\psi_i(\boldsymbol{0}) = \boldsymbol{x}_i$. The domain $\mathcal{P}_i$ will be taken as a subset of $T_{\boldsymbol{x}_i}\mathcal{M}$, and ψ_i is defined using the $m \times n$ matrix $\boldsymbol{\Psi}_i$, whose columns constitute an orthonormal basis of $T_{\boldsymbol{x}_i}\mathcal{M}$. This matrix sat-

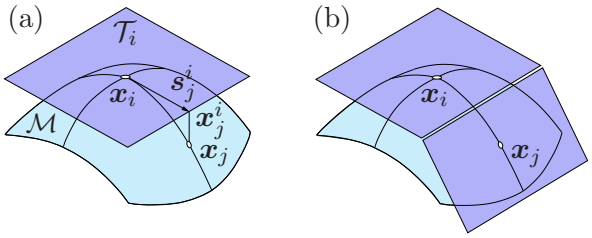


Figure 5: The higher-dimensional continuation method applied to a two-dimensional manifold in $\mathbb{R}^3$. Left: A point $\boldsymbol{x}_j \in \mathcal{M}$ can be obtained by projecting a point $\boldsymbol{x}_j^i = \boldsymbol{x}_i + \boldsymbol{s}_j^i$. Right: If a new chart is defined at $\boldsymbol{x}_j$, it must be properly coordinated with the chart at $\boldsymbol{x}_i$ so that their projections smoothly cover the manifold.

isfies

$$\left[\begin{array}{c} \boldsymbol{F}_{\boldsymbol{x}}(\boldsymbol{x}_i) \\ \boldsymbol{\Psi}_i^{\mathsf{T}} \end{array}\right] \boldsymbol{\Psi}_i = \left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{I}_{n\times n} \end{array}\right],$$

where $\boldsymbol{F}_{\boldsymbol{x}}$ is the Jacobian of $\boldsymbol{F}$. To compute the point $\boldsymbol{x}_j = \psi_i(\boldsymbol{s}_j^i) \in \mathcal{M}$ corresponding to a vector $\boldsymbol{s}_j^i \in \mathcal{P}_i$ we first obtain the point

$$\boldsymbol{x}_j^i = \boldsymbol{x}_i + \boldsymbol{\Psi}_i\, \boldsymbol{s}_j^i, \tag{16}$$

shown in Fig. 5(a), and project it down to $\mathcal{M}$ by solving

$$\left.\begin{array}{r} \boldsymbol{F}(\boldsymbol{x}_j) = \boldsymbol{0} \\ \boldsymbol{\Psi}_i^{\mathsf{T}}\,(\boldsymbol{x}_j - \boldsymbol{x}_j^i) = \boldsymbol{0} \end{array}\right\}$$

using a standard Newton-Raphson method initialized at $\boldsymbol{x}_j^i$ [38]. If the Newton method does not converge, $\boldsymbol{s}_j^i$ is assumed to be out of $\mathcal{P}_i$.

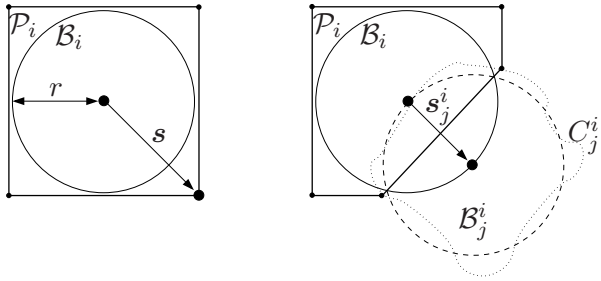


Figure 6: The process of chart construction. Left: The domain of chart C_i, $\mathcal{P}_i$, is initialized as a box enclosing a ball of radius r around x_i, both defined in the tangent space of $\mathcal{M}$ at x_i. Right: $\mathcal{P}_i$ is refined using a ball $\mathcal{B}_j^i$ that approximates C_j^i, the projection to C_i of the part of the manifold covered by the adjacent chart C_j.

4.2 Constructing an Atlas

Since the domain $\mathcal{P}_i$ of each chart C_i is limited, a full parameterization of the manifold requires the construction of a whole atlas. To this end, we use the fact that each point on the manifold is the potential center of a new chart, as shown in Fig. 5(b), and use the method in [24] to decide where to generate new chart centers and how to bound their associated domains. In this method, $\mathcal{P}_i$ is initialized as an n-dimensional hypercube enclosing a ball $\mathcal{B}_i$ of radius r, both defined in $T_{\boldsymbol{x}_i}\mathcal{M}$, and $\mathcal{P}_i$ is progressively clipped as new charts are added to the atlas (Fig. 6). The domain of each new chart C_j added to the atlas has to be properly coordinated with the domains of its neighboring charts in the atlas. As shown in Fig. 6, C_j is used to clip $\mathcal{P}_i$ using the intersection hyperplane between $\mathcal{B}_i$ and $\mathcal{B}_j^i$, a ball of radius r centered at the point given by $\boldsymbol{s}_j^i$ that approximates C_j^i, the projection on $T_{\boldsymbol{x}_i}\mathcal{M}$ of the part of the manifold covered by C_j. The procedure has mechanisms to adapt the area covered by each chart to the local curvature of the manifold.

When a chart C_i has been fully surrounded by other charts, $\mathcal{P}_i$ becomes a multifaceted polytope with all its vertices lying inside $\mathcal{B}_i$ and the chart is considered to be *closed*, meaning that no further expansion needs to be attempted from that chart. The process of chart expansion continues as long as there are open (non-closed) charts in the atlas. In the end, the connected component of $\mathcal{M}$ containing the initial point $\boldsymbol{x}_s$ gets fully covered by the generated charts.

As an example, Fig. 7 illustrates the progression of the algorithm on tracing a Chmutov surface from a given point, where open and closed charts are colored in red and blue, respectively. Since this is a two-dimensional manifold, here the balls $\mathcal{B}_i$ are circles, and the polytopes $\mathcal{P}_i$ are polygons.

Once an atlas has been computed, a graph G can be build whose nodes represent the chart centers, and whose edges correspond to the neighbouring relations between the generated charts. If a path connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}_g$ exists in $\mathcal{M}$, the chart covering $\boldsymbol{x}_g$ must be present in the atlas constructed from $\boldsymbol{x}_s$ and, thus, a solution path can be computed by searching G with standard graph search methods. If no chart containing $\boldsymbol{x}_g$ is found, path non-existence is established at the considered value of r.

4.3 Biasing the Search to the Goal

When multiple planning queries need to be resolved, it may be interesting to precompute a whole atlas of the connected component of $\mathcal{M}$ that is reachable from a given point. However, in single-query planning it is better to use

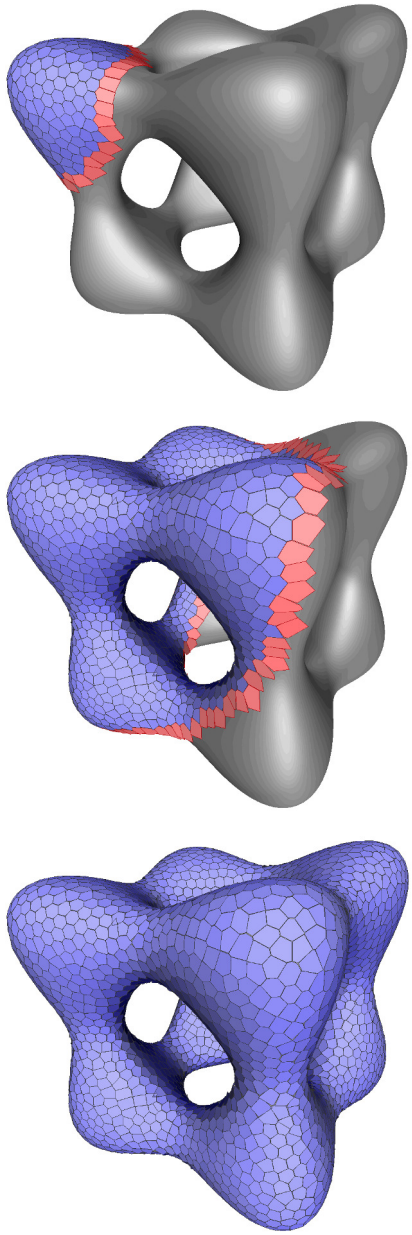


Figure 7: Progression of the continuation method on tracing a Chmutov surface defined by $3 + 8(x^4 + y^4 + z^4) = 8(x^2 + y^2 + z^2)$, using $r = 0.07$.

strategies that guide the expansion of the atlas towards $\boldsymbol{x}_g$ [35].

A possible approach is to use an A* search strategy [40], which computes a minimum-cost path from $\boldsymbol{x}_s$ to $\boldsymbol{x}_g$, assuming direct movements between adjacent chart centers. This approach provides a fair approximation of the shortest path on $\mathcal{M}$. At each iteration, this method expands the chart C_i with the lowest estimated cost of the whole movement from $\boldsymbol{x}_s$ to $\boldsymbol{x}_g$, while keeping a sorted priority queue of alternative path segments. The previous cost is the sum of a term $g(\boldsymbol{x}_i)$ that gives the lowest known cost of moving from $\boldsymbol{x}_s$ to the chart center $\boldsymbol{x}_i$, and a term $h(\boldsymbol{x}_i)$ that gives a lower bound of the cost of moving from $\boldsymbol{x}_i$ to $\boldsymbol{x}_g$. The value of the former term is maintained during the expansion of the atlas by means of a function $c(\boldsymbol{x}_j, \boldsymbol{x}_k)$ that determines the transition cost between two chart centers.

The A* strategy performs well in lower-dimensional C-spaces, but computation times considerably increase when n is large, due to the effort needed to find the optimal path. In such cases, one can simply use a Greedy Best-First strategy, in which the chart C_i to be expanded is just the one yielding a minimum estimated cost in the movement from $\boldsymbol{x}_i$ to $\boldsymbol{x}_g$. In contrast with the A* strategy, the Greedy Best-First strategy usually explores a smaller portion of the manifold. It does not necessarily generate all the neighbours of a chart under expansion, because such generation of children charts proceeds only until a chart with lower cost than the parent is found, and this stronly reduces the final number of charts produced. However, the path obtained by the Greedy Best-First method may be not be close to the optimal one. In both strategies, the search is stopped as soon as $\boldsymbol{x}_g$ is connected to the rest of the atlas, or when the full manifold has been covered without finding $\boldsymbol{x}_g$.

Since the returned paths use direct motions between adjacent chart centers, they may be slightly jerky. However, they can always be smoothed using standard path smoothing techniques [2]. Note also that any cost function can be used in principle. Depending on the particular context of application, the function may reflect energy consumption, travelled distance, or even a penalty due to robot collisions with itself or with the environment. In the latter case the function only has to assign an infinite cost to the chart-to-chart transitions that cause the collision [4].

5 Performance Tests

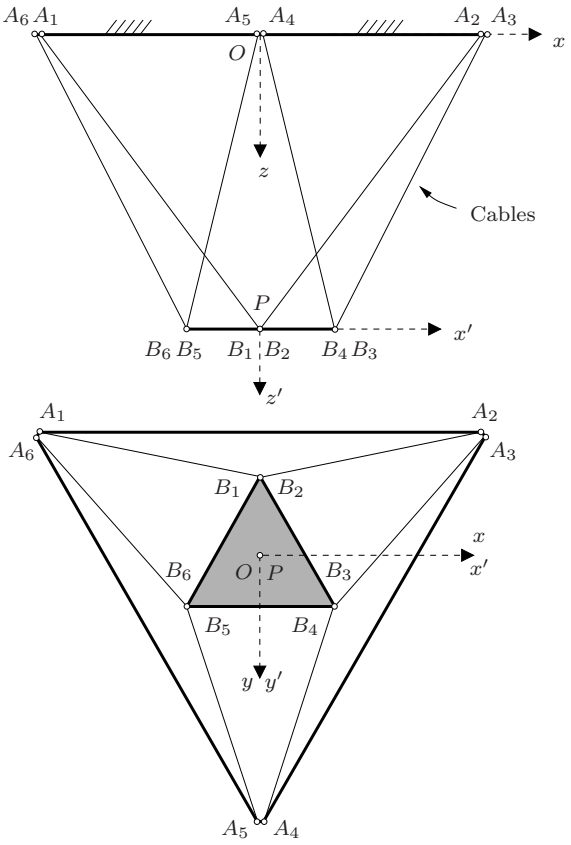
We next illustrate the performance of the method on two instances of an octahedral hexapod, specified as Robot 1 and 2 hereafter (Fig. 8). The robots essentially have the structure of the NIST Robocrane [1], but the planner remains applicable to general hexapods, with cable anchor points not necessarily coincident in pairs.

In Section 5.1 we apply the planner to compute paths in two-dimensional slices of $\mathcal{C}$ obtained by fixing four pose parameters. This shows how complex the wrench-feasible C-space can be even in simple cases, and stresses the advantages of our approach in comparison to previous methods based on discretization. Then, in Section 5.2, we use the planner in a real prototype, both to plan motions subject to geometric constraints, and free-flying motions in six-dimensional space.

Computation times are given in Table 1 at the end of the section. All results have been obtained with an implementation in C of the method available through [14], executed on a MacBook Pro computer equipped with a 2.66 GHz Intel Core i7 processor. Because of its attractive properties in parallel machines, the implementation adopts the tilt-and-torsion parameterization of $SO(3)$, for which $\boldsymbol{R} = \boldsymbol{R}_z(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\sigma - \phi)$, where ϕ, θ, and σ are the azimuth, tilt, and torsion angles respectively [7]. Thus, $\boldsymbol{\tau} = \{\phi, \theta, \sigma\}$ in this section, and the algorithms take into account that the angular coordinates differing in multiples of 2π refer to the same angle.

5.1 Planning in Illustrative Slices

In this example, Robot 1 is required to withstand a force of 1 N applied at a point P_m with position vector $\boldsymbol{p}_m = [30, 14, -21]^\mathsf{T}$ mm in frame $\mathcal{F}_2$. Note that the weight of this load corresponds to a constant wrench $\hat{\boldsymbol{w}}_0 = [0, 0, 1, 0, 0, 0]^\mathsf{T}$ (in SI units) if expressed in a frame $\mathcal{F}_3$ defined parallel to $\mathcal{F}_1$ and translating with P_m. The bounded perturbations of this wrench will be represented by the ellipsoid $\mathcal{K}$ centered in $\hat{\boldsymbol{w}}_0$ with $\boldsymbol{E} = 10^4\boldsymbol{I}_6$, also expressed in $\mathcal{F}_3$. Both $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ can be expressed



	Base	Platform
Robot 1	$A_1 = (-200, -115.47, 0)$	$B_1 = (0, -115.47, 0)$
	$A_2 = (200, -115.47, 0)$	$B_2 = (0, -115.47, 0)$
	$A_3 = (200, -115.47, 0)$	$B_3 = (100, 57.74, 0)$
	$A_4 = (0, 230.94, 0)$	$B_4 = (100, 57.74, 0)$
	$A_5 = (0, 230.94, 0)$	$B_5 = (-100, 57.74, 0)$
	$A_6 = (-200, -115.47, 0)$	$B_6 = (-100, 57.74, 0)$
Robot 2	$A_1 = (-231.62, -136.18, 0)$	$B_1 = (0, -89.15, 0)$
	$A_2 = (231.62, -136.18, 0)$	$B_2 = (0, -89.15, 0)$
	$A_3 = (233.74, -132.50, 0)$	$B_3 = (77.21, 44.57, 0)$
	$A_4 = (2.13, 268.67, 0)$	$B_4 = (77.21, 44.57, 0)$
	$A_5 = (-2.13, 268.67, 0)$	$B_5 = (-77.21, 44.57, 0)$
	$A_6 = (-233.74, -132.50, 0)$	$B_6 = (-77.21, 44.57, 0)$

Figure 8: Top: Front and top views of the octahedral architecture in a reference configuration. Bottom: Coordinates of A_i and B_i in mm, expressed in $\mathcal{F}_1 = Oxyz$ and $\mathcal{F}_2 = Px'y'z'$ respectively, for the two robots considered.

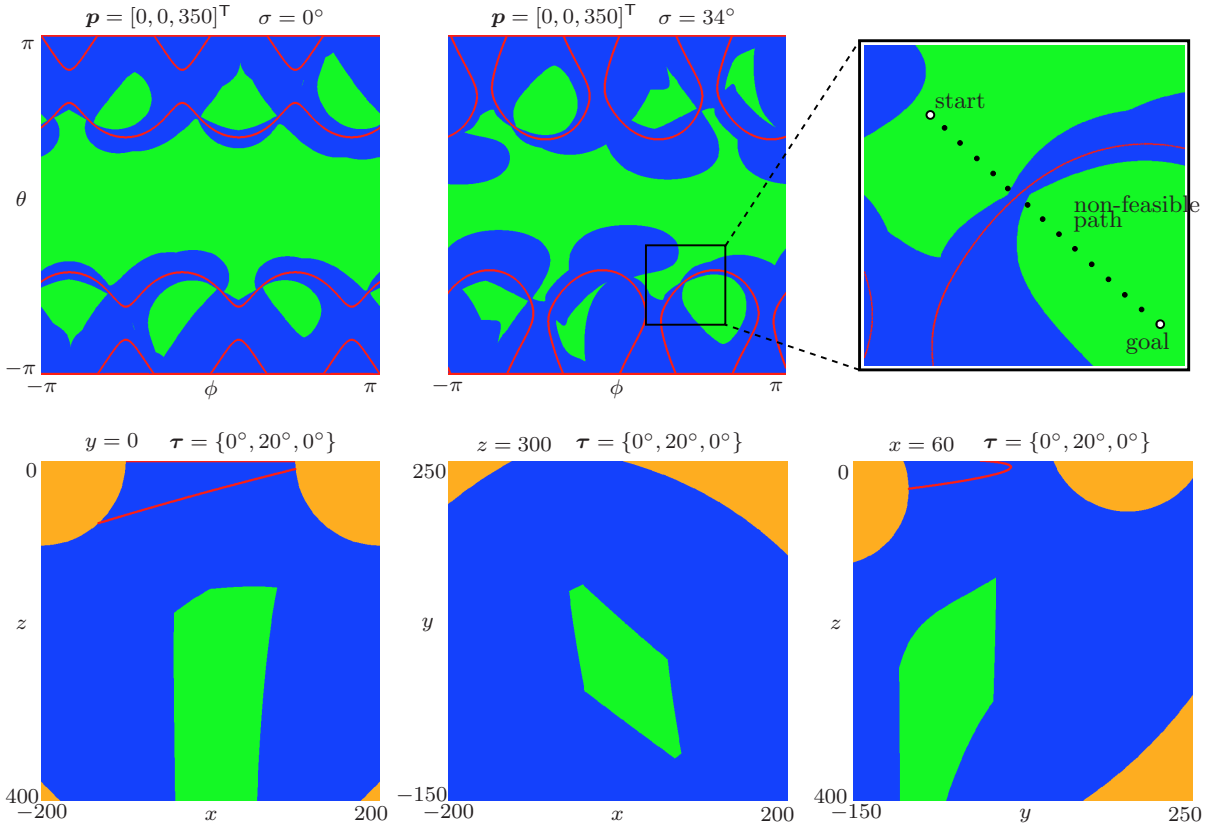


Figure 9: Slices of $\mathcal{C}$ for Robot 1 obtained by dense discretization. The images show the zones where some force is out of range (blue), some cable length exceeds its limits (orange), and configurations belonging to the wrench-feasible C-space (green). Note that the path in the top-right picture is unfeasible, despite the wrench-feasibility conditions are satisfied when evaluated at the shown points.

in $\mathcal{F}_1$ using Eqs. (20) and (21) of Appendix A. The tensions and lengths for all cables are constrained to the ranges $f_i \in (0.05, 0.5)$ N and $d_i \in (100, 550)$ mm.

Fig. 9 shows several slices of the wrench feasible C-space of the robot, computed in Matlab using dense discretization. Whereas the top row shows slices in which P and σ are held fixed, the bottom row shows others in which the full orientation $\boldsymbol{\tau}$ and one of the coordinates of P are constant. The configurations corresponding to $\mathcal{C}$ are indicated in green, while those that cannot be reached due to cable lengths or tensions out of range are represented by the orange and blue areas, respectively. The symmetries in the slices of the top row appear because $\{\phi, \theta, \sigma\}$ and $\{\phi + \pi, -\theta, \sigma\}$ represent the same orientation under the chosen parameterization. To avoid this double coverage of $SO(3)$, we only need to restrict the expansion of the atlas to the range $\theta \in [0, \pi]$. The figures also show in red the singularity curves where $det(\boldsymbol{J}(\boldsymbol{q})) = 0$, computed under no constraints on the cable tensions. It can be observed that, as expected, $\mathcal{C}$ naturally avoids crossing such curves, and that the navigation between two configurations is not trivial, because $\mathcal{C}$ is in general non-convex and may have very close connected components. The top-right plot of Fig. 9, in particular, exemplifies how evaluating the wrench-feasibility conditions at discrete points could result in erroneous paths linking configurations of $\mathcal{C}$ separated by singularity curves.

We apply our method to resolve a planning query on the top-centre slice of Fig. 9, hence

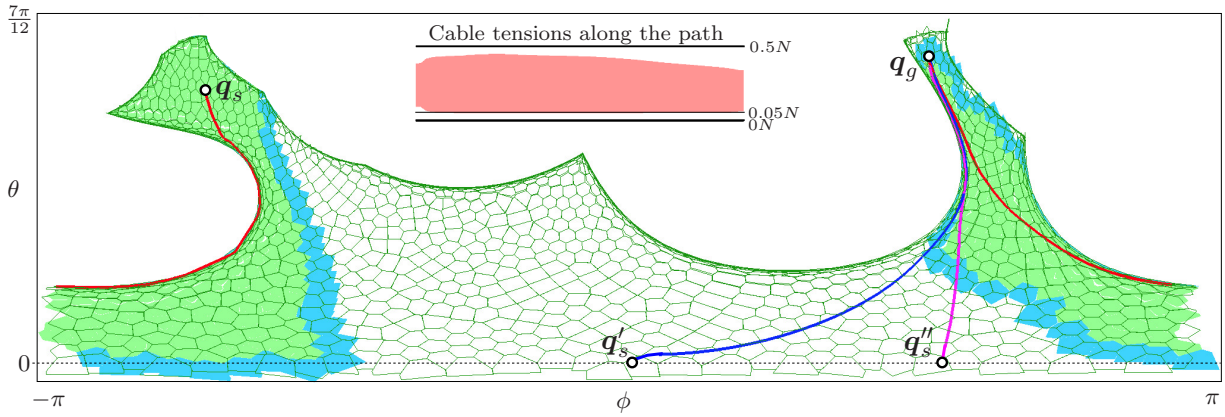


Figure 10: The three planning queries resolved in Section 5.1. The green mesh of hexagon-like polygons corresponds to the polytopes $\mathcal{P}_i$ of the whole atlas obtained from $\boldsymbol{q}_s$. Only part of such atlas is generated when trying to connect $\boldsymbol{q}_s$ to $\boldsymbol{q}_g$ under the A* strategy, shown shaded in green (closed charts) and blue (open charts). The inset at the top shows the envelope of cable tensions along this path. See the text for details.

exploring a manifold of dimension $n = 2$. Since here the platform can only rotate, we use the functions

$$c(\boldsymbol{x}_j, \boldsymbol{x}_k) = \|log(\boldsymbol{R}(\boldsymbol{\tau}_j)^\mathsf{T} \boldsymbol{R}(\boldsymbol{\tau}_k))\|,$$

and

$$h(\boldsymbol{x}_i) = c(\boldsymbol{x}_i, \boldsymbol{x}_g).$$

to implement the A* search strategy of the planner. For two orientations given by $\boldsymbol{R}(\boldsymbol{\tau}_j)$ and $\boldsymbol{R}(\boldsymbol{\tau}_k)$, $c(\boldsymbol{x}_j, \boldsymbol{x}_k)$ gives the angle of the axis-angle representation of $\boldsymbol{R}(\boldsymbol{\tau}_j)^\mathsf{T}\boldsymbol{R}(\boldsymbol{\tau}_j)$, which is a standard metric of $SO(3)$ [28]. The start and goal configurations $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ are given by the $\boldsymbol{\tau}$ values $\{-2.3, 1.6, \frac{17}{90}\pi\}$ rad and $\{1.7, 1.7, \frac{17}{90}\pi\}$ rad, and by the position of P, which is fixed at $\boldsymbol{p} = [0, 0, 350]^\mathsf{T}$ mm.

Fig. 10 shows the path returned by the planner, in red, after smoothing it. The green mesh corresponds to the full atlas of the connected component of $\mathcal{C}$ attainable from $\boldsymbol{q}_s$, and the shaded area indicates the part of this component that was actually explored to obtain the path using the A* search scheme. Green and blue shading corresponds to closed and open charts respectively, and it can be seen that the algorithm takes into account the topology of the angle variables, allowing to traverse from $\phi = -\pi$ to $\phi = \pi$, or *vice versa*.

Observe that a naive approach based on interpolating the start and goal configurations would violate some of the constraints of $\mathcal{C}$, giving rise to uncontrollable motions or unaffordable cable tensions. The computed path, in contrast, correctly avoids these situations and guarantees control of the platform at all points, while keeping cable lengths and tensions within their limits. Fig. 10, for example, shows that the envelope of cable forces is admissible along the movement. Furthermore, because we rely on continuation, the synthesized path will not misleadingly bridge two disjoint components of $\mathcal{C}$, a property that cannot be ensured by discretization-based strategies.

The line $\theta = 0$ of Fig. 10 is known to be a representation singularity because all of its points correspond to a same orientation [7]. To illustrate that this is not a problem in practice, we show the results of planning two movements starting at distinct points of the $\theta = 0$ line, $\boldsymbol{q}'_s$ and $\boldsymbol{q}''_s$, both leading to $\boldsymbol{q}_g$. The resulting paths, shown in blue and purple, are clearly different, because the nature of the algorithm does not capture the fact that there is no cost of moving between two configurations with $\theta = 0$. However, $\mathcal{M}$ is smooth despite the singularity, and the planner has no problem in computing feasible movements in the two cases. An alter-

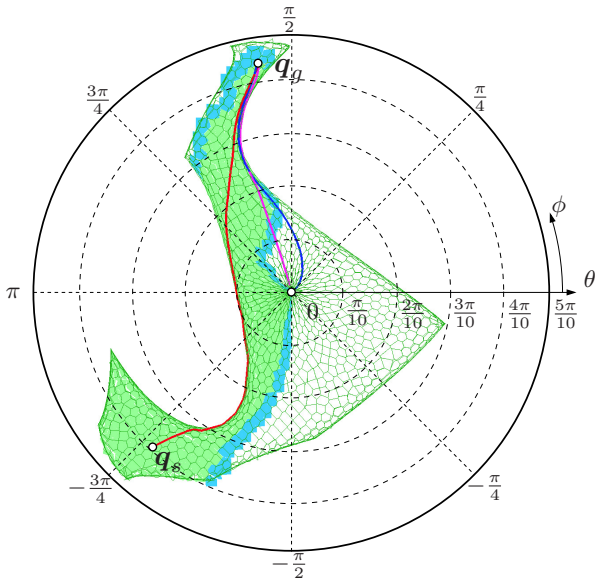


Figure 11: Polar representation of Fig. 10. The blue and purple paths connect the same two poses of the platform, but are computed using different initial values of the angle ϕ.

native way to represent these results is by using the cylindrical coordinate system proposed by [7], in which the line $\theta = 0$ corresponds to the center point (Fig. 11). In this projection there is a bijection between the $\{\phi, \theta\}$ pairs and their corresponding orientations, and we can see that the blue and purple paths are similar, only differing in the vicinity of $\theta = 0$.

5.2 Planning in a Real Prototype

In order to mimic a situation in which the platform is subject to geometric constraints, we next apply our approach to the robot of Fig. 12, which is meant to perform insertion tasks on the surface of a sphere. For operation purposes, the platform is required to move tangentially to the sphere, with zero torsion. Using the parametric form of Eq. (14), these conditions can be written as follows

$$\left.\begin{array}{l} \boldsymbol{p} = \boldsymbol{r}_c + r_s \begin{bmatrix} \cos\alpha_2 \cos\alpha_1 \\ \cos\alpha_2 \sin\alpha_1 \\ -\sin\alpha_2 \end{bmatrix} \\ \phi = \alpha_1 - \pi \\ \theta = \frac{\pi}{2} - \alpha_2 \\ \sigma = 0 \end{array}\right\}, \qquad (17)$$

where $\boldsymbol{r}_c = [x_s, y_s, z_s]^\mathsf{T}$ and r_s indicate the sphere center and radius, and α_1 and α_2 are two angular parameters. Thus, $\boldsymbol{\lambda} = \{\alpha_1, \alpha_2\}$ in this case, and the navigation manifold is of dimension $n = 2$ after adding Eq. (17) to Eq. (13).

The points A_i and B_i correspond to those of Robot 2 in Fig. 8, and the sphere is of radius 100 mm, with its center located at $\boldsymbol{r}_c = [0, 0, 306]^\mathsf{T}$ mm in frame $\mathcal{F}_1$. However, since a small distance between the platform and the sphere needs to be kept, a value of $r_s = 130$ mm is used in Eq. (17). The platform weight is of 0.6 kg, with its center of mass located in P, and we use the same matrix $\boldsymbol{E} = 10^4 \boldsymbol{I}_6$ as before. Cable tensions are limited by the maximum force allowable by the motors, with $f_i \in (0.1, 6.58)$ N, and the feasible lengths are those satisfying $d_i \in (200, 600)$ mm, for $i = 1, \ldots, 3$.

The resulting C-space is shown in Fig. 13 projected on the sphere, using the same drawing conventions of Fig. 10. The initial configuration, and the configurations where the insertion tasks are to be done are referred to as $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, and $\boldsymbol{q}_3$, respectively, and correspond to the $\boldsymbol{\lambda}$ values $\{0.55, \frac{\pi}{2}\}$, $\{0.55, 0.75\}$, and $\{2.63, 0.75\}$. If we ask the planner to synthesize movements from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$, and then to $\boldsymbol{q}_3$, we obtain the red path in Fig. 13, which has been computed using

$$c(\boldsymbol{x}_i, \boldsymbol{x}_j) = r_s \arctan\left(\frac{\|\boldsymbol{n}_i \times \boldsymbol{n}_j\|}{\boldsymbol{n}_i \cdot \boldsymbol{n}_j}\right),$$

and

$$h(\boldsymbol{x}_i) = c(\boldsymbol{x}_i, \boldsymbol{x}_g), \qquad (18)$$

in the A* search strategy, where $\boldsymbol{n}_i = \boldsymbol{p}_i - \boldsymbol{r}_c$. Given two points $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ on the sphere, these functions provide the great-circle distance between them, so the algorithm returns motions that minimize the distance travelled by P on the surface of the sphere. A simple planning approach based on interpolation in the $\{\alpha_1, \alpha_2\}$ plane would result in a rather different motion. The transition from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$ would coincide, but the movement from $\boldsymbol{q}_2$ to $\boldsymbol{q}_3$ would yield the blue path of the figure, which rapidly leaves $\mathcal{C}$ at the beginning.

The video in `http://youtu.be/DM9lPROmMKY` shows the results of executing both the interpolated and planned versions

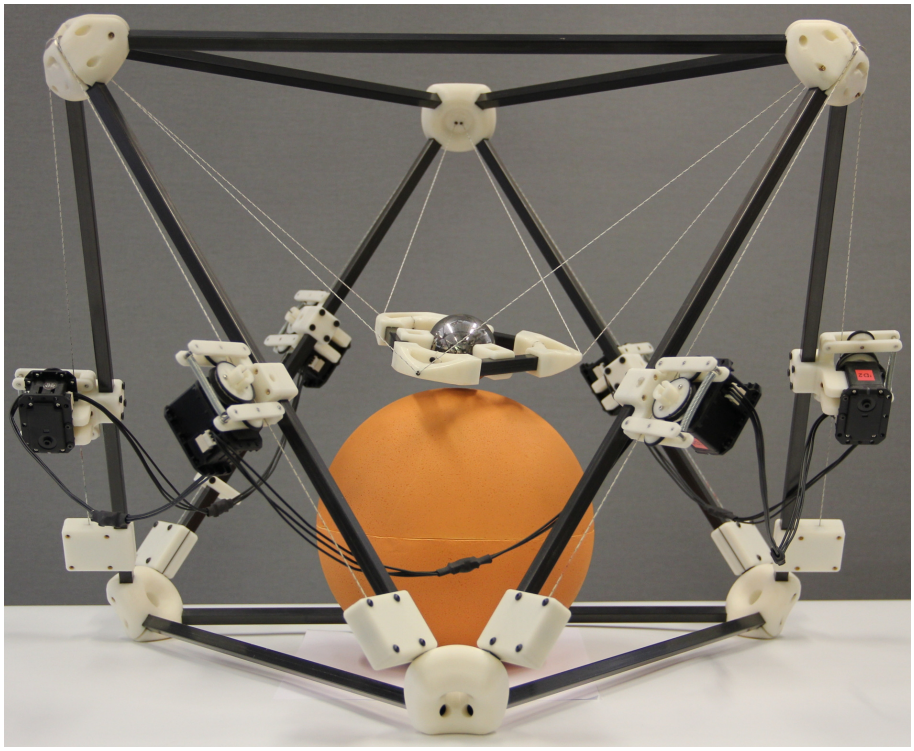


Figure 12: A cable-driven hexapod constructed at Institut de Robòtica i Informàtica Industrial (IRI) to demonstrate the planning method proposed.

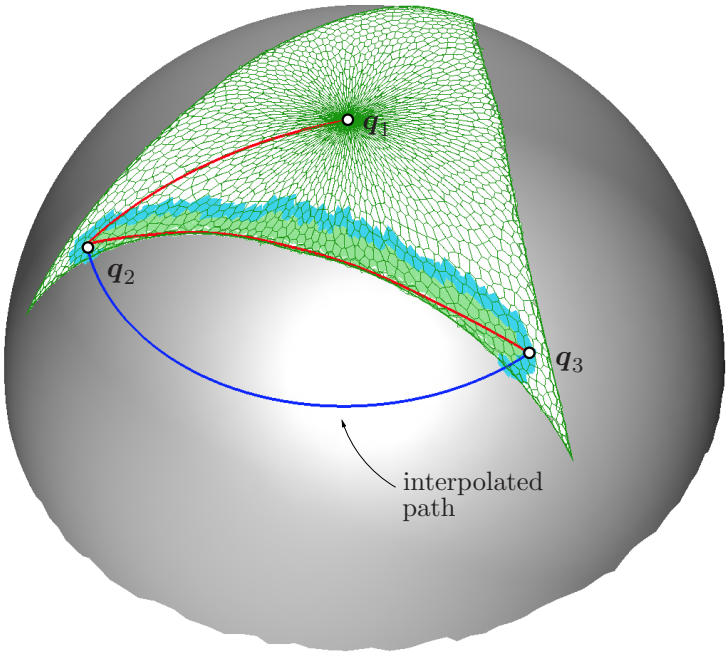


Figure 13: Results of planning a path from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$, and then to $\boldsymbol{q}_3$, in the first experiment of Section 5.2. The part of the C-space explored to plan the transition from $\boldsymbol{q}_2$ to $\boldsymbol{q}_3$ is shown shaded in green.

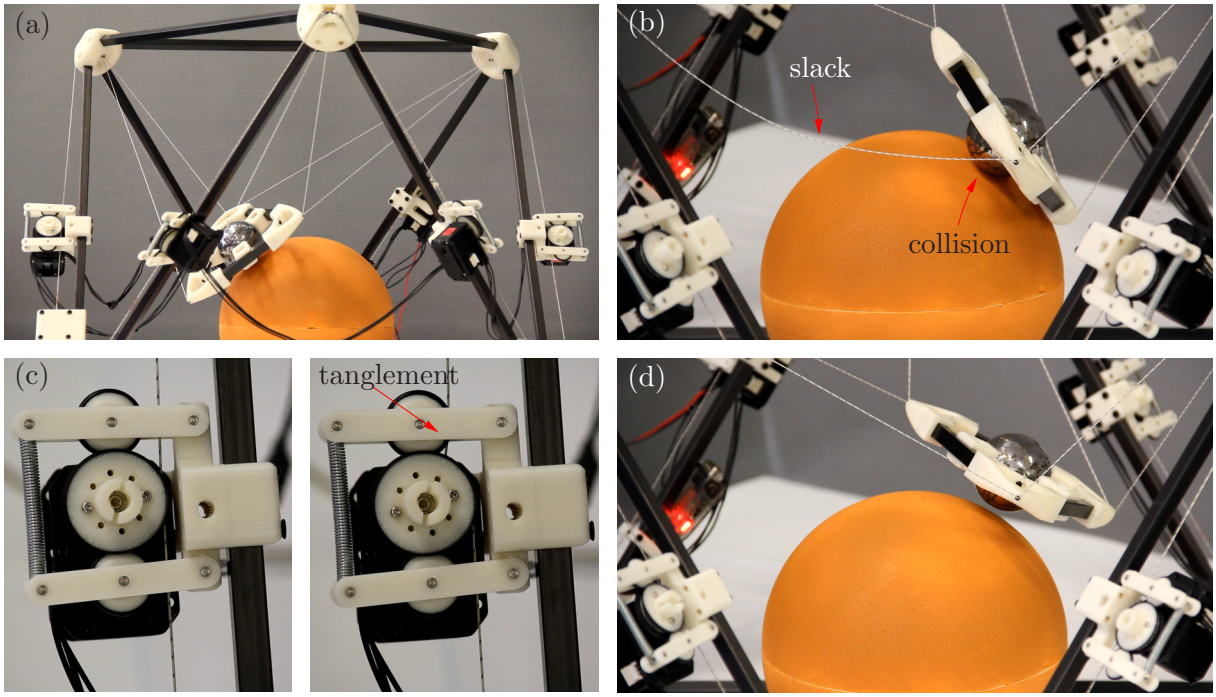


Figure 14: Several snapshots of the video available in `http://youtu.be/DM9lPROmMKY`. (a): The robot at configuration $\boldsymbol{q}_2$. (b): Slackness of the cables and collisions appear when trying to follow the interpolated path from $\boldsymbol{q}_2$ to $\boldsymbol{q}_3$. (c): Comparison between normal operation (left) and tanglement of the cables in the motors (right). (d): Following the wrench-feasible path between $\boldsymbol{q}_2$ and $\boldsymbol{q}_3$ guarantees a smooth motion and control of the platform at all times.

of the $\boldsymbol{q}_1 - \boldsymbol{q}_2 - \boldsymbol{q}_3$ movement. It can be seen how, as expected, the platform moves smoothly from $\boldsymbol{q}_1$ to $\boldsymbol{q}_2$, but some cables become slack and control of the platform is lost along the interpolated path from $\boldsymbol{q}_2$ to $\boldsymbol{q}_3$. Other undesirable effects of following this path include shakyness of the platform under small perturbations, collisions with the environment, and cable tanglement at the motors. In contrast, control of the platform is maintained when following the path $\boldsymbol{q}_2 - \boldsymbol{q}_3$ returned by the planner. Fig. 14 summarizes the experiment in a few snapshots.

The method can be applied to higher-dimensional problems as well. For example, if the insertion operations are to be performed with an axisymmetric tool, we can ignore the zero-torsion constraint on the platform pose by removing $\sigma = 0$ in Eq. (17). The result is a three-dimensional planning problem that is efficiently solved with the method, although the computation time is higher due to the increased size of the search space. Six-dimensional problems can also be solved by taking only Eq. (13) into account. Assuming that the sphere is no longer present, for example, a free-flying motion from $\boldsymbol{q}_2$ to $\boldsymbol{q}_3$ can be planned in a matter of seconds using the Greeedy Best-first strategy.

The problem sizes and computation times of all test cases are reported in Table 1. For each case, the table shows the dimension of the explored manifold (n), the number of problem variables involved (m), and the time spent by the Greedy Best-first and A* strategies (last two columns expressed in seconds), using the same cost functions. It becomes apparent how, in terms of computation time, the use of a Greedy Best-first strategy is advantageous in higher-dimensional problems, while the A* one is affordable and adviseable in lower dimensions, because it normally yields lower-cost

paths. Moreover, it must be said that once a partial atlas has been computed, all planning queries between configurations in such atlas can be solved in a few milliseconds.

Robot	Path	n/m	GBF	A*
1	$\boldsymbol{q}_s \rightarrow \boldsymbol{q}_g$	2/150	29	62
	$\boldsymbol{q}'_s \rightarrow \boldsymbol{q}_g$	2/150	6	12
	$\boldsymbol{q}''_s \rightarrow \boldsymbol{q}_g$	2/150	5	9
2	$\boldsymbol{q}_1 \rightarrow \boldsymbol{q}_2$	2/155	12	2
	$\boldsymbol{q}_2 \rightarrow \boldsymbol{q}_3$	2/155	30	14
	$\boldsymbol{q}_2 \rightarrow \boldsymbol{q}_3$ (σ free)	3/155	56	166
	$\boldsymbol{q}_2 \rightarrow \boldsymbol{q}_3$ (free-flying)	6/153	40	> 6000

Table 1: Problem sizes and computation times (in seconds) for all test cases.

6 Conclusions

The ability to govern a load both in position and orientation is crucial in many applications, and parallel cable-driven robots constitute an advantageous, cost-effective solution. The problem is challenging though, because cable tensions need to be positive to avoid swaying and unwanted collisions of the load when moving towards a goal. This paper has proposed a path planner that ensures a safe navigation in this context. The planner automatically computes motion paths that, at any point, allow to counteract a platform wrench subject to bounded perturbations, with cable tensions lying inside their allowable bounds. When executing the planned paths, the load moves smoothly and predictably towards the desired pose, thus making the approach suitable in fine manipulation tasks especially. As shown in the paper, moreover, the strategy naturally avoids crossing the forward singularity locus, and the adjustment of the tension bounds gives a meaningful way to tune the clearance relative to such locus. Although the emphasis has been on modelling the length and tension constraints of the hexapod, the algorithm is flexible-enough to also accommodate collision constraints of the robot. As shown in Section 4, these simply translate into infinite cost transitions in the graph of the atlas. The method has been thoroughly tested in C-spaces of various dimensions, and with experiments on a real prototype. Video sequences of the latter can be found in the multimedia material attached to the paper.

A number of points are proposed for future attention. First, in some applications it may be necessary to also obtain platform motions with a certain degree of position accuracy for the moving load. Due to the generality of the numerical continuation strategy, it shouldn't be difficult to deal with such constraints using developments analogous to those in the paper, propagating known bounds on the position error of the actuators to an ellipsoidal bound on the platform pose. Second, while inertia effects can currently be modelled as bounded perturbations of the wrench, this approach is better suited to move the robot quasi-statically, as shown in the experiments performed. Further research needs to be done to see whether the method can be extended to also synthesize motion paths ensuring a full dynamic control of the robot, or even a time-optimal trajectory. As explained in [21], substantial workspace enlargements might be achieved in doing so. Finally, efforts should also be made to try to generalise the method to deal with cable-driven robots with more than six cables, which, despite their more intrusive nature, are increasingly proposed to exploit their redundant actuation. Here, the difficulty resides in that the mapping of the wrench ellipsoid to the cable-tension ellipsoid is not easily expressed in closed form, because a same wrench can be counteracted by infinite cable tension combinations. Careful analysis of tension distribution methods might give a clue to this extension [36].

A Obtaining the Wrench Ellipsoid

To obtain the center $\hat{\boldsymbol{w}}_0$ and shape matrix $\boldsymbol{E}$ needed in Section 2, we distinguish two situations, depending on whether the suspended load is moving in free space, or it is subject to

a contact constraint.

In the former case (Fig. 15 (a)) the resultant wrench $\hat{\boldsymbol{w}}$ is the sum of the load weight $\hat{\boldsymbol{w}}_w$ and small perturbations introduced by inertia forces or external agents like the wind. Both $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ are easier to describe in a frame $\mathcal{F}_3$ parallel to $\mathcal{F}_1$ in this case, located in the center of mass G of the load. In this frame $\boldsymbol{E}$ is a constant matrix, and

$$\hat{\boldsymbol{w}}_0 = \hat{\boldsymbol{w}}_w = [0, 0, -w, 0, 0, 0]^\mathsf{T}, \qquad (19)$$

where w is the load weight. However, since in Section 2 and Eq. (4) $\boldsymbol{J}$ was given in $\mathcal{F}_1$, both $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ need to be converted into such frame for consistency. To convert $\hat{\boldsymbol{w}}_0$ we can use the transformation

$$\hat{\boldsymbol{w}}^i = \boldsymbol{e}_{ij} \cdot \hat{\boldsymbol{w}}^j, \qquad (20)$$

where $\hat{\boldsymbol{w}}^i$ and $\hat{\boldsymbol{w}}^j$ indicate the coordinates of a same wrench in frames $\mathcal{F}_i$ and $\mathcal{F}_j$, and

$$\boldsymbol{e}_{ij} = \left[\begin{array}{c|c} \boldsymbol{R}_{ij} & \boldsymbol{0}_3 \\ \hline \boldsymbol{X}_{ij}\boldsymbol{R}_{ij} & \boldsymbol{R}_{ij} \end{array}\right]$$

is the change of reference matrix [18]. The block $\boldsymbol{R}_{ij}$ is the rotation matrix providing the orientation of $\mathcal{F}_j$ relative to $\mathcal{F}_i$, and

$$\boldsymbol{X_{ij}} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix},$$

where x, y, and z are the coordinates of the origin of $\mathcal{F}_j$ in frame $\mathcal{F}_i$. In Fig. 15 (a), e.g., if $\boldsymbol{g}$ is the position vector of G in frame $\mathcal{F}_2$ we have

$$[x, y, z]^\mathsf{T} = \boldsymbol{p} + \boldsymbol{R}\,\boldsymbol{g},$$

where $\boldsymbol{p}$ and $\boldsymbol{R}$ are defined as in Section 2. Similarly, to convert $\boldsymbol{E}$, it is easy to see that if $\boldsymbol{E}^i$ and $\boldsymbol{E}^j$ refer to the shape matrix of an ellipsoid in frames $\mathcal{F}_i$ and $\mathcal{F}_j$, then

$$\boldsymbol{E}^i = \boldsymbol{e}_{ji}^\mathsf{T} \cdot \boldsymbol{E}^j \cdot \boldsymbol{e}_{ji}, \qquad (21)$$

where

$$\boldsymbol{e}_{ji} = \left[\begin{array}{c|c} \boldsymbol{R}_{ij}^\mathsf{T} & \boldsymbol{0}_3 \\ \hline \boldsymbol{R}_{ij}^\mathsf{T}\boldsymbol{X}_{ij}^\mathsf{T} & \boldsymbol{R}_{ij}^\mathsf{T} \end{array}\right].$$

In contact situations (Fig. 15 (b)), the resultant wrench $\hat{\boldsymbol{w}}$ is the sum of the weight $\hat{\boldsymbol{w}}_w$, the contact wrench $\hat{\boldsymbol{w}}_c$, and bounded perturbations in both of such wrenches. We thus think of $\hat{\boldsymbol{w}}$ as the sum of two vectors lying inside six-dimensional ellipsoids $\mathcal{K}_w$ and $\mathcal{K}_c$ respectively, centered in $\hat{\boldsymbol{w}}_w$ and $\hat{\boldsymbol{w}}_c$, and with shape matrices $\boldsymbol{E}_w$ and $\boldsymbol{E}_c$. Therefore, $\hat{\boldsymbol{w}}$ will lie inside the Minkowski sum of $\mathcal{K}_w$ and $\mathcal{K}_c$, which, according to [39], can be tightly bounded by an ellipsoid centered in

$$\hat{\boldsymbol{w}}_0 = \hat{\boldsymbol{w}}_w + \hat{\boldsymbol{w}}_c, \qquad (22)$$

with shape matrix

$$\boldsymbol{E} = \frac{1}{2}\boldsymbol{E}_w(\boldsymbol{E}_w + \boldsymbol{E}_c)^{-1}\boldsymbol{E}_c. \qquad (23)$$

Assuming that $\hat{\boldsymbol{w}}_w$, E_w, $\hat{\boldsymbol{w}}_c$, and E_c have all been expressed in frame $\mathcal{F}_1$, these are the expressions needed for $\hat{\boldsymbol{w}}_0$ and $\boldsymbol{E}$ in Section 2.

B Proofs of the Properties

B.1 Non-singularity of the Screw Jacobian

It is easy to prove that $\boldsymbol{J}$ is full rank for all $\boldsymbol{q} \in \mathcal{C}$ by contradiction. Let us assume that there is a configuration $\boldsymbol{q} \in \mathcal{C}$ for which $\boldsymbol{J}$ is rank deficient. Then, since $\boldsymbol{B}$ would be rank deficient as well, we have $\dim(\ker \boldsymbol{B}) \geq 1$. In this case, there would be some i for which $\ker \boldsymbol{B} = \ker \boldsymbol{B}^i$, i.e. all solutions $\boldsymbol{v}_i$ of $\boldsymbol{B}^i\boldsymbol{v}_i = \boldsymbol{0}$ would lie in $\ker \boldsymbol{B}$. This would imply $\boldsymbol{v}_i^\mathsf{T}\boldsymbol{B}\boldsymbol{v}_i = 0$ for such i, which contradicts Eq. (5). Therefore, $\boldsymbol{J}$ must be non-singular for all $\boldsymbol{q} \in \mathcal{C}$.

B.2 Smoothness of the Navigation Manifold

To prove that $\mathcal{M}$ is a smooth manifold, it is sufficient to show that $\boldsymbol{F}(\boldsymbol{x})$ is differentiable and has a full rank Jacobian $\boldsymbol{F}_{\boldsymbol{x}}$ for all $\boldsymbol{x} \in \mathcal{M}$. Then, the smoothness of $\mathcal{M}$ follows from the Implicit Function Theorem.

By construction, all functions intervening in $\boldsymbol{F}(\boldsymbol{x})$ are clearly differentiable. After reordering the equations, $\boldsymbol{F}_{\boldsymbol{x}}$ can be expressed in the

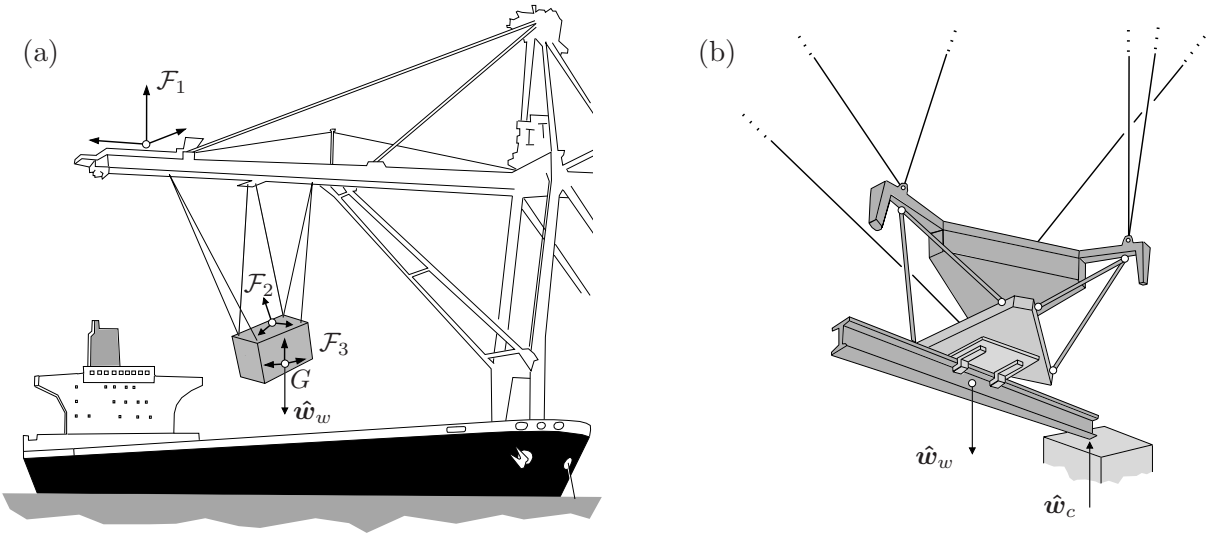


Figure 15: Obtaining the wrench ellispsoid in two common scenarios. (a) Cargo retrieval [1]. (b) Beam mounting using the NIST gripper shown in Fig. 1 [32].

following block-triangular form

$$\boldsymbol{F_x} = \begin{bmatrix} \boldsymbol{\Phi_y} & & & & & \\ * & \boldsymbol{J} & & & & \\ * & & \begin{matrix} 2\boldsymbol{v}_1^\mathsf{T}\boldsymbol{B} \\ \boldsymbol{B}^1 \end{matrix} & & & \\ & & \ddots & & & \\ & & & \begin{matrix} 2\boldsymbol{v}_6^\mathsf{T}\boldsymbol{B} \\ \boldsymbol{B}^6 \end{matrix} & & \\ & * & * & & \boldsymbol{S} & \\ & * & * & & & \boldsymbol{T} \end{bmatrix},$$

where the empty blocks represent zero-matrices and the asterisks indicate non-zero blocks. The block $\boldsymbol{\Phi_y}$ is of size 30×36, and the remaining diagonal blocks are all of size 6×6. To see that $\boldsymbol{F_x}$ is full rank, we next check that all the blocks in its diagonal are full-rank:

1. The block $\boldsymbol{\Phi_y}$ is the Jacobian of the subsystem $\boldsymbol{\Phi}(\boldsymbol{y}) = \boldsymbol{0}$ formed by Eqs. (1), (2), and (10), where

   $$\boldsymbol{y} = [\boldsymbol{p}, \boldsymbol{\tau}, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_6, d_1, \ldots, d_6, g_1, \ldots, g_6]. \tag{24}$$

   This Jacobian takes the form

   $$\boldsymbol{\Phi_y} = \begin{bmatrix} * & * & -\boldsymbol{I}_{18} & & \\ & & * & \boldsymbol{L} & \\ & & & * & \boldsymbol{G} \end{bmatrix}, \tag{25}$$

   where the block columns correspond to the derivatives of the variables in $\boldsymbol{y}$ in the order of Eq. (24), and $\boldsymbol{L}$ and $\boldsymbol{G}$ are 6×6 diagonal matrices with $-2d_i$ and $(d_i - \underline{d_i}) \cdot (\overline{d_i} - d_i)$ in their entries, respectively. Observe that $\boldsymbol{\Phi_y}$ is full rank, because the entries in $\boldsymbol{L}$ and $\boldsymbol{G}$ do not vanish over $\mathcal{M}$, by virtue of Eq. (10) and the fact that $\underline{d_i} > 0$.

2. Appendix B.1 proves that the block $\boldsymbol{J}$ is full rank for all $\boldsymbol{q} \in \mathcal{C}$, and the same argument shows that it must be so for all $\boldsymbol{x} \in \mathcal{M}$.

3. The 6×6 block matrices involving $\boldsymbol{B}$ and $\boldsymbol{B}^i$ can only be rank deficient if $v_{i,i} = 0$ but, as explained in Section 3.2, this can never happen on $\mathcal{M}$.

4. Finally, $\boldsymbol{S}$ and $\boldsymbol{T}$ are 6×6 diagonal matrices whose elements are $(f_{0,i} - v_{i,i} - \underline{f_i})$ and $(\overline{f_i} - f_{0,i} - v_{i,i})$, respectively, which do not vanish because of Eqs. (11) and (12).

B.3 Smoothness of Lower-dimensional Subsets

Subsets of $\mathcal{M}$ defined by adding Eq. (14) or Eq. (15) to Eq. (13) are also smooth manifolds.

When adding Eq. (14), the Jacobian of the resulting system of equations takes the form

$$\left[\begin{array}{c|c|c|c} \boldsymbol{\Omega_\lambda} & \boldsymbol{I}_6 & & \\ \hline & \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \\ \hline & * & * & \boldsymbol{P} \end{array}\right], \tag{26}$$

where $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ are the matrices formed by the first two and last three block-columns of $\boldsymbol{\Phi_y}$ in Eq. (25), respectively, and $\boldsymbol{P}$ is the submatrix of $\boldsymbol{F_x}$ obtained by removing its first block-row and block-column. The Jacobian in (26) is full rank, because the square blocks $\boldsymbol{I}_6$, $\boldsymbol{\Phi}_2$ and $\boldsymbol{P}$ are all full rank. Similarly, if we add Eq. (15) to Eq. (13) the resulting Jacobian is

$$\left[\begin{array}{c|c|c} \boldsymbol{C_{p,\tau}} & & \\ \hline \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \\ \hline * & * & \boldsymbol{P} \end{array}\right], \tag{27}$$

which is clearly full rank because $\boldsymbol{C_{p,\tau}}$ is non-singular according to our assumptions.

References

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

[2] Dmitry Berenson, Siddhartha S Srinivasa, and James Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1460, Oct 2011.

[3] S. Bhattacharya, H. Hatwal, and A. Ghosh. Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators. *Mechanism and Machine Theory*, 33(7):965–974, 1998.

[4] O. Bohigas, M.E. Henderson, L. Ros, M. Manubens, and J.M. Porta. Planning singularity-free paths on closed-chain manipulators. *Robotics, IEEE Trans. on*, 29(4):888–898, Aug 2013.

[5] O. Bohigas, M. Manubens, and L. Ros. Navigating the wrench-feasible C-space of cable-driven hexapods. In T. Bruckmann and A. Pott, editors, *Cable-Driven Parallel Robots*, pages 53–68. Springer, 2012.

[6] Oriol Bohigas. *Numerical Computation and Avoidance of Manipulator Singularities*. PhD thesis, Universitat Politècnica de Catalunya, 2013. Available through the web address `http://goo.gl/wlR0i`.

[7] I. A. Bonev, D. Zlatanov, and C. M. Gosselin. Advantages of the modified Euler angles in the design and control of PKMs. In *Proceedings of the 3rd Chemnitz Parallel Kinematics Seminar/2002 Parallel Kinematic Machines International Conference, Chemnitz, Germany*, pages 171–188, 2002.

[8] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser. NIMS-PL: A cable-driven robot with self-calibration capabilities. *IEEE Trans. on Robotics*, 25(5):1005 –1015, 2009.

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

[10] P. Bosscher, R. L. Williams II, L. S. Bryson, and D. Castro-Lacouture. Cable-suspended robotic contour crafting system. *Automation in Construction*, 17:45–55, 2007.

[11] Roger Bostelman, James Albus, Nicholas Dagalakis, Adam Jacoff, and John Gross. Applications of the NIST RoboCrane. In *Proceedings of the 5th International Symposium on Robotics and Manufacturing*, pages 14–18, 1994.

[12] P. Cheng, J. Fink, S. Kim, and V. Kumar. Cooperative towing with multiple robots. In Gregory Chirikjian, Howie Choset, Marco Morales, and Todd Murphey, editors, *Algorithmic Foundations of Robotics VIII*, volume 57 of *Springer Tracts in Advanced Robotics*, pages 101–116. Springer Berlin / Heidelberg, 2009.

[13] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer, 2011.

[14] CUIK Project home page. `http://www-iri.upc.es/groups/gmr/cuikweb`, 2013.

[15] Nicholas G Dagalakis, James S Albus, B-L Wang, Joseph Unger, and James D Lee. Stiffness study of a parallel link robot crane for shipbuilding applications. *Journal of Offshore Mechanics and Arctic Engineering*, 111(3):183–193, 1989.

[16] Bhaskar Dasgupta and T.S. Mruthyunjaya. Singularity-free path planning for the Stewart platform manipulator. *Mechanism and Machine Theory*, 33(6):711–725, 1998.

[17] Anjan Kumar Dash, I-Ming Chen, Song Huat Yeo, and Guilin Yang. Workspace generation and planning singularity-free path for parallel manipulators. *Mechanism and Machine Theory*, 40(7):776–805, 2005.

[18] J.K. Davidson and K.H. Hunt. *Robots and Screw Theory: Applications of Kinematics and Statics to Robotics*. Oxford University Press, 2004.

[19] I. Ebert-Uphoff and P.A. Voglewede. On the connections between cable-driven robots, parallel robots and grasping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4521–4526, 2004.

[20] S. Fang, D. Franitza, R. Verhoeven, and M. Hiller. Optimum motion planning for tendon-based Stewart platforms. In H. Tian, editor, *Proc. of the 11th IFToMM World Congress in Mechanism and Machine Science*, Tianjin, China, 2003. China Machinery Press.

[21] Clement Gosselin, Ping Ren, and S. Foucault. Dynamic trajectory planning of a two-DOF cable-suspended parallel robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1476–1481, May 2012.

[22] M. Gouttefarde, D. Daney, and J.-P. Merlet. Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots. *IEEE Trans. on Robotics*, 27(1):1–13, 2011.

[23] T. C. Harmon, R. F. Ambrose, R. M. Gilbert, J. C. Fisher, M. J. Stealey, and W. J. Kaiser. High-resolution river hydraulic and water quality characterization using rapidly deployable networked infomechanical systems (NIMS RD). *Environmental Engineering Science*, 24(2):151–159, 2007.

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

[25] M. E. Henderson. Multiparameter parallel search branch switching. *International Journal of Bifurcation and Chaos in Applied Science and Engineering*, 15(3):967–974, 2005.

[26] M. E. Henderson. *Numerical Continuation Methods for Dynamical Systems*, chapter Higher-Dimensional Continuation, pages 77–115. Springer, 2007.

[27] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Franitza. Design, analysis and realization of tendon-based parallel manipulators. *Mechanism and Machine Theory*, 40(4):429–445, 2005.

[28] Du Q Huynh. Metrics for 3D rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[29] S. Lahouar, E. Ottaviano, S. Zeghoul, L. Romdhane, and M. Ceccarelli. Collision free path-planning for cable-driven parallel robots. *Robotics and Autonomous Systems*, 57(11):1083–1093, 2009.

[30] John M Lee. *Introduction to Smooth manifolds*. Springer Verlag, New York, 2001.

[31] A. M. Lytle, K. S. Saidi, R. V. Bostelman, W. C. Stone, and N. A. Scott. Adapting a teleoperated device for autonomous control using three-dimensional positioning sensors: experiences with the NIST

RoboCrane. *Automation in Construction*, 13:101–118, 2004.

[32] Alan M Lytle and Kamel S Saidi. NIST research in autonomous construction. *Autonomous Robots*, 22(3):211–221, 2007.

[33] J.-P. Merlet and D. Daney. A portable, modular parallel wire crane for rescue operations. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2834 –2839, 2010.

[34] S. R. Oh, J. C. Ryu, and S. K. Agrawal. Dynamics and control of a helicopter carrying a payload using a cable-suspended robot. *Journal of Mechanical Design*, 128(5):1113–1121, 2006.

[35] J. M. Porta, L. Jaillet, and Oriol Bohigas. Randomized path planning on manifolds based on higher-dimensional continuation. *The International Journal of Robotics Research*, 31(2):201–215, 2012.

[36] Andreas Pott. An improved force distribution algorithm for over-constrained cable-driven parallel robots. In *Computational Kinematics*, pages 139–146. Springer, 2014.

[37] J. Pusey, A. Fattah, S. K. Agrawal, and E. Messina. Design and workspace analysis of a 6-6 cable-suspended parallel robot. *Mechanism and Machine Theory*, 3:761–778, 2004.

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

[39] Lluís Ros, Assumpta Sabater, and Federico Thomas. An ellipsoidal calculus based on propagation and fusion. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 32(4):430–442, 2002.

[40] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[41] Kamel Saidi, Geraldine Cheok, Marek Franaszek, Christopher Brown, Jeremy Swerdlow, Robert Lipman, Itai Katz, Mani Golparvar-Fard, Paul Goodrum, Manu Akula, Gabriel Dadi, and Behshad Ghadimi. Development and use of the NIST Intelligent and Automated Construction Job Site Testbed. Technical report, National Institute of Standards and Technology, 2011.

[42] Shamik Sen, Bhaskar Dasgupta, and Asok Kumar Mallik. Variational approach for singularity-free path-planning of parallel manipulators. *Mechanism and Machine Theory*, 38(11):1165–1183, 2003.

[43] E. Stump and V. Kumar. Workspaces of cable-actuated parallel manipulators. *ASME Journal of Mechanical Design*, 128(1):159–167, 2006.

[44] S. Tadokoro, T. Matsushima, Y. Murao, and H. Kohkawa. A parallel cable-driven motion base for virtual acceleration. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1700 –1705, 2001.

[45] R. Verhoeven. *Analysis of the workspace of tendon based Stewart platforms*. PhD thesis, Universität Duisburg-Essen, 2004.

[46] R. Verhoeven, M. Hiller, and S. Tadokoro. Workspace of tendon-driven Stewart platforms: Basics, classification, details on the planar 2-DOF class. In *Proc. of the 4th International Conference on Motion and Vibration Control*, pages 871–876, 1998.

[47] P.A. Voglewede and I. Ebert-Uphoff. Overarching framework for measuring closeness to singularities of parallel manipulators. *IEEE Trans. on Robotics*, 21(6):1037–1045, 2005.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy under project DPI2014-57220-C2-2-P, by the CSIC project 201250E026, and by a Juan de la Cierva fellowship supporting Montserrat Manubens.
The authors also wish to express their gratitude to M. E. Henderson for introducing them to higher-dimensional continuation, to Patrick Grosch for constructing the robot prototype used in the paper, and to Josep M. Porta for his valuable work on developing the CUIK suite.

IRI reports

This report is in the series of IRI technical reports. All IRI technical reports are available for download at the IRI website
`http://www.iri.upc.edu`.