The forward kinematics of cable-driven parallel robots with sagging cables

J-P. Merlet and J. Alexandre-dit-Sandretto

Abstract Solving the forward kinematics (FK) of parallel robots is known to be a difficult task and the problem is even more complex for cable driven parallel robot (CDPR): the system of equations that has to be solved is larger than with rigid legs as first the static equations have to be taken into account and, second because the deformation of the cables because of their elasticity and their mass may play a role, while being described by a relatively non algebraic complex model. We consider in this paper any arbitrary CDPR whose cables may present a significant deformation due to their elasticity and own mass and we present for the first time an interval analysis based generic algorithm that allows to calculate in a guaranteed manner all the FK solutions and illustrate its use for a CDPR with 8 cables.

1 Introduction

Cable-driven parallel robot (CDPR) have the mechanical structure of the Gough platform with rigid legs except that the legs are cables whose length may be controlled. We will assume that the output of the coiling system for cable i is a single point A_i, while the cable is connected at point B_i on the platform (figure 1). Classical kinematics problem are the inverse kinematics (find the lengths of the cables for a given pose of the platform) and the forward kinematics (FK) (find the pose(s) of the platform for given cable lengths). Solving the FK of parallel robots is one of the most challenging problem in modern kinematics. Two categories of FK may be distinguished:

J-P. Merlet
INRIA, 2004 Route des Lucioles, 06902 Sophia-Antipolis Cedex, e-mail: Jean-Pierre.Merlet@inria.fr

J. Alexandre-dit-Sandretto
INRIA, 2004 Route des Lucioles, 06902 Sophia-Antipolis Cedex, e-mail: julien.alexandre-dit-sandretto@ensta-paristech.fr
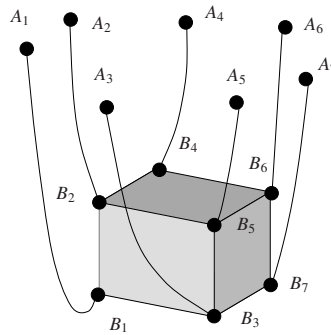
Fig. 1 Cable driven parallel robots with sagging cables

- *real-time FK*: used for control purposes with the objective of determining the current pose of the robot. It need to be fast (running within a sampling time of the robot controller) and uses the knowledge that the solution is "close" to a known pose. It may also be used for simulation purposes
- *full FK*: the purpose is to determine all solutions of the FK. It is used off-line for determining the possible initial states of the robot when running a simulation

We will see later on that the real time FK is not a problem and we will address only the full FK. This problem has been addressed for CDPRDs in a very preliminary stage only very recently and under restrictive assumptions on the behavior of the cables. Indeed we may consider:

- *non-deformable cables*: they are aligned along the direction A_iB_i
- *elastic cables*: they are also aligned along A_iB_i but their lengths depend upon the tension to which they are submitted
- *catenary cables*: they exhibit elasticity and their own mass leads to a deformed shape

Most of the kinematic works have assumed non-deformable cables. For robots having at least 6 cables the FK is equivalent to the one of classical parallel robots (for more than 6 cables at most 6 will be under tension simultaneously [1]). With less than 6 cables the problem is still open as the geometrical constraints relating the length of the cables to the pose leads to less equations than unknowns, which imposes to add the 6 additional statics equations and the cable tensions as additional unknowns. For a CDPR with m cables the minimal system has $6+m$ equations in $6+m$ unknowns, to be compared with the system of 6 equations for the Gough platform. Although there has been progress recently to solve these problems [2, 3, 4] there are still a lot of progress to be made in order to determine the maximal number of solutions according to m, solutions that should have only positive tensions and are stable.

If we assume elasticity in the cables there has been some works for the IK [5, 6] but to the best of our knowledge the FK has never been addressed in the general case.

This paper addresses the most general case of FK for CDPR having catenary type cables. Very few works have addressed the IK and FK of such a robot [7, 8, 9] and none have considered the general case.

2 Problem statement

2.1 Cable model

We will assume cables with linear density μ, cross-section A_0 and E will denote the Young modulus of the cable material. A reference frame $O, \mathbf{x}, \mathbf{y}, \mathbf{z}$ will be used and the coordinates of the A_i points are known in this frame. In the vertical plane of the cable we may assume that the cable is attached at point A with coordinates $(0,0)$ while the other extremity is attached at point B with coordinates (x_b, y_b) (figure 2). The vertical and horizontal forces F_z, F_x are exerted on the cable at point B and the cable length at rest is L_0. With this notation the coordinates of B are related to the
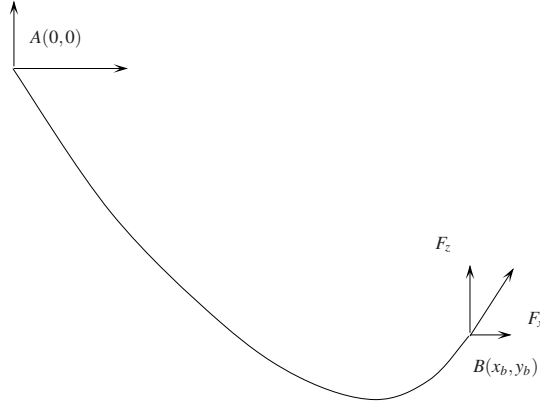

Fig. 2 A deformed cable

forces F_x, F_z [10] by:

$$x_b = F_x \left(\frac{L_0}{EA_0} + \frac{sinh^{-1}(F_z) - sinh^{-1}((F_z - \frac{\mu g L_0}{F_x})}{\mu g} \right) \tag{1}$$

$$z_b = \frac{\sqrt{F_x^2 + F_z^2} - \sqrt{F_x^2 + (F_z - \mu g L_0)^2}}{\mu g}$$
$$+ \frac{F_z L_0}{EA_0} - \frac{\mu g L_0^2}{2EA_0} \tag{2}$$

2.2 The FK problem

We consider a spatial robot with m cables whose lengths will be denoted $L_0^1, \ldots, L_0^m$. Without loss of generality we will assume that the Young modulus, linear density and cross-section of the cables are identical. The problem we have to solve is to determine all the possible poses of the platform being given the L_0^j and the location of the A_i, together with the external forces/torques $\mathscr{F}$ that act on the platform. In terms of unknowns we will assume a minimal representation of the pose with 6 parameters and we have also the $2m$ unknown forces F_x^i, F_z^i, for a total of $6 + 2m$ unknowns. Note that to express these forces in the reference frame we need to establish a rotation matrix **Rot** that rotates the planar frame around the z axis by an angle u_i so that in the reference frame we get the force acting on the platform **F** as $\mathbf{F} = \mathbf{Rot}(-F_x, 0, -F_z)^T$ and hence for cable j $\mathbf{F_j} = (-F_x^j \cos u_j, F_x^j \sin u_j, F_z^j)$. If we assume that the external force acting on the platform is the gravity and that the platform mass is M the mechanical equilibrium imposes that:

$$\sum_{j=1}^{j=m} F_x^j \cos u_j = 0 \quad \sum_{j=1}^{j=m} F_x^j \sin u_j = 0 \quad \sum_{j=1}^{j=m} F_z^j = Mg \tag{3}$$

If C is the center of mass of the platform we get also

$$\sum_{j=1}^{j=m} \mathbf{CB_j} \times \mathbf{F_j} = \mathbf{0} \tag{4}$$

In terms of equations we have the $2m$ equations (1),(2) and the 6 static equations that express the mechanical equilibrium of the platform. Hence we end up with $6 + 2m$ equations so that solving the FK requires to solve a square system, which will usually have a finite number of solutions. It may also been seen that the FK in that case is much more complex than the FK of the Gough platform (it has $2m$ additional equations) and that the classical methods used to determine an upper bound of the maximum number of solutions (Bezout number, elimination, Gröbner basis) cannot be applied here as equations (1,2) are not algebraic.

3 Solving the FK

As a theoretical solving appears to be difficult to be used we will have to resort to a numerical solving method, that has to provide all the solutions. We will use an interval analysis (IA) approach, which guarantees to find all the solutions lithing withing some given ranges. The basis of IA is the *interval evaluation*: being given a function $f(x_1, \ldots, x_n)$ in n variables and assuming that each variable x_i lies in the range $[\underline{x_i}, \overline{x_i}]$ the interval evaluation of f is a range $[A, B]$ such that $\forall x_i \in [\underline{x_i}, \overline{x_i}], i \in [1, n]$ we have $A \le f(x_1, \ldots, x_n) \le B$. There are multiple ways to define an interval eval-

uation but the most simple is the *natural evaluation*: each mathematical operator has an interval equivalent (for example the addition interval operator $\pm$ is defined as $[a,b]\pm[c,d] = [a+c,b+d]$) and transforming any function by using the interval operators allows to calculate the interval evaluation. One of the property of the interval evaluation $[A,B]$ is that if $A > 0$ of $B < 0$, then f cannot cancel whatever is the value of the variables in their ranges. Note that an interval evaluation may be *overestimated*: there may not be value of the variables in their respective range such that $f(x_1,\ldots,x_n) = A$ or $f(x_1,\ldots,x_n) = B$. Indeed overestimation may occur because of multiple occurrences of a given variable that are considered as independent: for example the evaluation of $x - x$ when $x \in [-1,1]$ is [-2,2]. But the size of the overestimation decreases with the width of the variable ranges.

The second key ingredient of IA is the *branch and bound* algorithm. A *box* $\mathscr{B}$ is defined as a set of ranges for the variables. If for a given box we have $f(\mathscr{B} = [A,B]$ with $A < 0, B > 0$, then we select one of the variable, x_i bisect its range in two and create two new boxes $\mathscr{B}_1, \mathscr{B}_2$ that are identical to $\mathscr{B}$ except for the range for x_i which result from the bisection. These boxes are stored in a list and will be processed later on. We will see in the next section that if a box is tiny enough we may determine if it includes a single solution and compute this solution with an arbitrary accuracy.

However we will not use the minimal set of equations for the FK. Indeed the pose of the platform will not be represented by the coordinates of C and three orientation angles. The motivation is that coordinates x_b, z_b in (1, 2) will be obtained after using the rotation matrix and will include several occurrences of the rotation angles, possibly leading to large overestimation. We prefer to represent the pose of the platform by the 12 coordinates in the reference frame of 4 of the points B_i, which are not coplanar (we assume here that B_1, B_2, B_3, B_4 are chosen). With this choice the coordinates of any point M on the platform may be obtained as $\mathbf{OM} = \alpha_1 \mathbf{OB_1} + \alpha_2 \mathbf{OB_2} + \alpha_3 \mathbf{OB_3} + \alpha_4 \mathbf{OB_4}$ where the α are known constants. Such a representation allows one to obtain the coordinates of the $B_j, j > 4$ and of the center of mass C. As we have now 12 unknowns for representing the pose of the platform instead of 6 with the minimal representation we need 6 additional equations that are obtained by stating that the distance between a pair of points in the set $B_1, \ldots B_4$ is a known constant. Note that these equations are not sufficient to fully describe the geometry of the platform (e.g the equations does not allow to differentiate if a B_j point is over or under the plane that includes the three other points). Another test is needed and we will use the fact that for any point M of the platform there exist constants β_j such that

$$\mathbf{B_jM} = \beta_1 \mathbf{B_jB_k} + \beta_2 \mathbf{B_jB_l} + \beta_3 \mathbf{B_jB_k} \times \mathbf{B_jB_l}, \quad j,k,l \in [1,4] \qquad (5)$$

It remains to manage the angles u_i: formally they can be obtained using the equation

$$x_b^j \sin(u_j) + y_b^j \cos(u_j) = 0 \qquad (6)$$

where x_b^j, y_b^j are the coordinates of B_j in the reference frame. But by so doing their interval evaluation will be relatively large even for small width for the x_b^j, y_b^j inter-

vals, especially if $j > 4$. Hence we prefer to add them as additional unknowns and to use (6) as additional equations. Hence we end up with $12 + 3m$ unknowns for $12 + 3m$ equations. Note that we have checked that solving this system in the context of a real-time FK is not difficult as soon as a certified strategy is used [11]: this strategy allows one to determine the current pose of the robot or eventually that the pose is too close to a singularity (in which case the FK has an infinite number of solutions).

An IA approach impose to determine a domain in which are located all the solutions. This can be easily done for the x_b^j, y_b^j variable that are restricted to lie within the convex hull of the A_i points. This can also be done for z_b^j, the z coordinate of B_j that cannot be greater than the highest z coordinate of the A_i points and cannot be lower than the length obtained if we assume that the cable is vertical and bears the platform. As for the u_i as the cable have to lie within the convex hull of the A_i we can also get bounds for these variables. It remains the variables F_x, F_z which have no natural bounds except that F_x cannot be negative. We will first define m new variables λ_j such that $F_z^j = \lambda_j F_x$ (which allows one to have simpler expressions for (1, 2). We take then as upper bound for F_x 10 times the value of mg and for the λ a range of [-10,10] (at the extremities of this range the cables are almost vertical).

3.1 Determining exact solutions

The classical IA branch and bound algorithm assume that if the width of a box is smaller than a small value and the interval evaluations of all equations include 0, then we have found an approximate solution of the system. Here we proceed in another way: for each box of the algorithm we run a few iterations of the Newton-Raphson scheme with as estimate of the solution the center of the box H. Note that even if the NR algorithm converges there is no guarantee 1) that the result is indeed an approximate solution of the system, 2) that the solution lie within the box or even within the search space, 3) that the result satisfies the constraint (5). In order to check if the result is really a solution of the system we use Kantorovitch theorem [11] that allows one to verify that there is indeed a single solution of the system in a ball centered at H with a known radius. If this test succeed we have furthermore the property that the NR scheme, initialized with H as guess point, converges toward the solution. We will see in the implementation section that this property will allow us to compute an approximation of the solution with an arbitrary accuracy.

As soon as a solution H_0 is found it is stored and in a first step we will assume that there is no other solution in a ball centered at H_0 with a given radius, this being applied only on the 12 coordinates of the point $B_1, \ldots, B_4$. Any box that is fully included in this ball will be eliminated and if a box has an intersection with the ball, then the intersection part will be removed from the box. Our purpose in this first step is to determine balls that include a solution and possibly others. In a second step we will run the algorithm on this ball and this check will be faster because the

search domain will be drastically reduced. With this approach the IA algorithm is guaranteed to complete.

3.2 Heuristics

A drawback of the usual IA branch and bound algorithm that eliminates boxes only according to the interval evaluation of the equations is that is is not efficient as soon as we have complex equations with multiple occurrences of the variables. But several heuristics allows one to drastically improve the efficiency of the algorithm. A first set of heuristics is called the *consistency* approach, which is based on a rewriting of the equations. Consider for example the equation that described that the distance between the points of a pair of B_i point is constant. This equation is written as

$$(x_b^i - x_b^j)^2 + (y_b^i - y_b^j)^2 + (z_b^i - z_b^j)^2 = d_{ij}^2 \tag{7}$$

which may be rewritten as

$$(x_b^i - x_b^j)^2 = d_{ij}^2 - (y_b^i - y_b^j)^2 - (z_b^i - z_b^j)^2$$

Let $[A, B]$ denote the interval evaluation of the right hand-side of this equation. We deduce that

- if $B < 0$ the equation has no solution
- if $A \leq 0, B \geq 0$ then $-\sqrt{B} + x_b^j \leq x_b^i \leq -x_b^j + \sqrt{B}$
- if $A \geq 0$ then x_b^i belongs to $[-\sqrt{B} + x_b^j, -\sqrt{A} + x_b^j] \cup [\sqrt{A} + x_b^j, \sqrt{B} + x_b^j]$

With this approach we may improve the range for any variable in the equation or even eliminate a box without having to use the bisection process. It is important to note that if the set of variable is denoted x and we are able to write an equation under the form $g(x_i) = G(x)$ the consistency requires an inverse operator of g in order to be able to update x_i. This also motivate our choice not to use the minimal representation of the pose but a more algebraic formulation whose inverse is trivial. In our implementation the consistency is applied on all equations of the system and for all variables. It is also used on equations (5) and on the equations $\mathbf{B_i B_j} . \mathbf{B_i B_k} = d_{ij} d_{ik} \cos(\theta)$ where θ is the known angle between the lines going through (B_i, B_j) and (B_i, B_k).

Another efficient heuristic is the *3B method*. Assume that we have a box and select one of the variable x_i whose range is $[\underline{x_i}, \overline{x_i}]$. We change this range to $[\underline{x_i}, [\underline{x_i} + \varepsilon]$, where ε has a small value. Interval evaluation of the equations and the consistency are used to determine if this new box may include a solution. If the answer is negative we can safely modify the initial box by setting the range for x_i to $[\underline{x_i} + \varepsilon, \overline{x_i}]$. This process is applied for all variables but also on the right side of the interval for x_i.

Another approach is used for the equations that have multiple occurrences of the same variable. We calculate the gradient of these equations and it interval evaluation

for the current box. If this evaluation has a constant sign we set the variable to the appropriate lower or upper bound of the range to improve the interval evaluation. This process has to be recursive: as soon as a variable is set to an extremity of its range, then the interval evaluation of the gradient for another variable may become of constant sign.

Another important issue is the choice of the variable that will be selected for the bisection process. Our strategy is to bisect in priority the 12 coordinates of the $B_1, \ldots B_4$ until the width of their interval is lower than a given threshold. Indeed these variables play an important role in the equations: if they are fixed the equations (1, 2) admit a single solution that correspond to the minimal potential energy of the cables.

3.3 Implementation

The algorithm is implemented using the interval arithmetics of BIAS/PROFIL[1] while the higher level uses the functions of our `ALIAS` library[2] that mixes a C++ library and a Maple interface. The Maple interface has allowed to generate most of the C++ code for the algorithm and includes an arbitrary accuracy Newton scheme which allow us to calculate an approximation of the solution with n digits, the n-th digit being guaranteed to be exact, n being a number given by the end-user.

Another property of the Maple interface is that it allows one to implement the algorithm in a *distributed manner*, i.e. running the algorithm on several computers. Indeed it must be noticed that in the solving scheme the treatment of a given box is independent from the treatment of the other boxes to be processed. This allow to have a master program that manages the list of boxes to be processed and the list o solutions and an arbitrary number m of slave computers. The master computer process the initial box until it has a fixed number of boxes in its list. Then it sends the top boxes to the slave computer that a few iterations of the solving algorithm and send back to the master the eventual solution and the boxes that remain to be treated. As the communication overhead is small compared to the computing time of the algorithm the distributed version allows to divide the processing time by m

4 Example

We consider the large scale robot developed by LIRMM and Tecnalia as part of the ANR project Cogiro [7] (figure 3). This robot is a suspended CDPR (i.e. there is no cable pulling the platform downward) with 8 cables, whose A_i coordinates are given in table 1. The cables characteristics are $E = 100^9 N/m^2$, $\mu = 0.346$ kg/m and

[1] http://www.ti3.tuhh.de/keil/profil/index_e.html

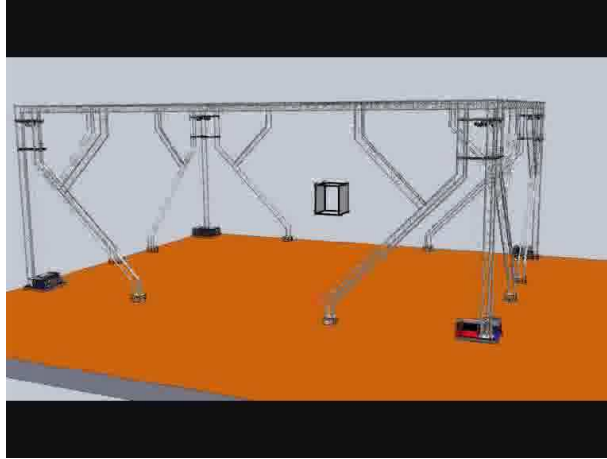[2] http://www-sop.inria.fr/coprin/developpements/main.html

Fig. 3 The robot developed for the the ANR Cogiro project. Although the robot is real we present a CAD drawing that allows one to better figure out the CDPR.

x	y	z	x	y	z
-7.175	-5.244	5.462	-7.316	-5.1	5.47
-7.3	5.2	5.476	-7.161	5.3	5.485
7.182	5.3	5.488	7.323	5.2	5.499
7.3	-5.1	5.489	7.161	-5.27	5.497

Table 1 Coordinates of the attachment points on the base (in meters)

their diameter is 10 mm. The mass of the platform of 10 kg. The value of the L_0s (table 2) are the non-deformed cable lengths for the pose (1,0,2) in meters and for an orientation such that the reference frame and the mobile frame are aligned.

1	2	3	4
10.48215	9.838952	10.16035	8.96827
5	6	7	8
10.310003	8.421629	8.663245	8.655556

Table 2 Lengths of the cable at rest (in meters)

With 8 cables we have to solve a system of 36 equations and this is probably the most challenging FK task that has even been considered. The solving algorithm has been implemented using 10 computers and nineteen solutions were found in the search domain in a computation of about 24 hours. They are presented in table 3, while the cable tensions are given in table 4. The solutions are depicted in figure 4. It is interesting to note that the solution poses are distributed all over the possible

workspace: for example the x, y, z coordinates of B_1 are in the ranges [0.29,1.86], [-0.74,0.98], [2.37,4.99]. The F_x forces exhibit also a very large range. For example for cable 1 this force ranges from 20.08N to 417N. We observe the same variation for the F_z force: for the same cable its ranges from -30.25N to 15.91 N. In 15 cases on 19 the F_Z tension in cable 1 is positive, meaning that the cable exert a downward force on the platform. The number of cables that exert an upward force to support the load is either 2, 3 or 4, meaning that only a small subset of cables contributes to this support: this may be an useful information for dimensioning the cable.

5 Conclusions

We have presented for the first time a generic algorithm to solve the FK for CDPR with sagging cables. This a computer intensive algorithm (because of the complexity of the problem), that is however guaranteed to provide all solutions. A test case of a robot with 8 cables (probably one of the most complex that has been studied) has shown that we may obtain surprising poses. As prospective our objective is to determine a better balance between the various heuristics that are used in the solving. We will also study the stability of the solutions, possibly introducing stability condition as an additional solving heuristic in order to speed up the computation.

Acknowledgment

Sol	x_1	y_1	z_1	x_2	y_2	z_2	x_3	y_3	z_3
1	1.55	0.43	4.26	0.04	0.21	4.89	1.05	-0.35	4.71
2	0.57	-0.08	3.53	1.47	-0.80	4.71	0.44	-0.89	4.15
3	1.45	-0.47	2.37	0.42	0.34	3.39	0.43	-0.27	2.38
4	1.27	0.65	4.41	-0.04	-0.31	4.11	1.10	-0.37	4.39
5	0.37	0.47	3.96	1.27	-0.33	5.10	1.34	0.11	4.01
6	0.91	0.98	3.45	0.94	0.04	4.81	1.64	0.64	4.09
7	1.53	-0.74	3.40	0.47	0.53	3.49	1.13	-0.18	4.16
8	1.46	-0.42	4.29	0.52	0.33	3.16	0.94	0.46	4.25
9	1.06	0.64	4.06	1.00	-0.60	2.97	0.93	-0.38	4.12
10	1.66	-0.07	3.45	0.46	0.77	4.20	0.94	-0.30	4.16
11	0.71	0.64	4.10	1.08	-0.64	3.11	0.58	-0.39	4.14
12	1.65	-0.32	4.99	0.73	-0.07	3.64	0.83	-0.78	4.57
13	0.29	0.69	3.64	1.11	-0.67	3.17	0.12	-0.32	3.71
14	1.39	0.45	3.81	0.55	-0.23	2.55	0.88	0.81	2.99
15	1.28	0.53	3.78	0.11	0.01	4.82	1.22	-0.20	4.50
16	1.04	0.90	3.63	0.54	-0.12	4.82	1.52	0.33	4.34
17	1.37	0.52	3.59	0.19	-0.64	3.63	0.44	0.40	3.15
18	0.71	0.92	3.69	0.53	-0.69	4.01	-0.08	0.31	3.94
19	1.86	-0.15	4.12	0.27	-0.60	4.19	1.18	-0.31	4.88

Table 3 The coordinates of $B1, B_2, B_3$ for the 19 solutions (meter)

1 F_x	417.00	22.12	78.64	421.40	14.78	17.25	48.67	98.43
1 F_z	-30.25	15.05	11.19	-43.41	14.86	12.21	4.61	12.48
2 F_x	20.08	138.03	48.65	160.06	292.98	26.46	361.66	11.28
2 F_z	12.79	5.90	10.45	3.60	-16.58	7.61	-38.66	11.63
3 F_x	53.41	51.34	48.02	57.21	55.95	56.22	53.79	61.07
3 F_z	0.49	5.01	1.03	-6.48	5.28	-0.82	-6.66	-1.12
4 F_x	63.89	17.85	429.43	41.10	14.32	73.80	129.88	305.89
4 F_z	11.08	13.29	-29.11	6.18	14.03	5.90	-13.06	-11.58
5 F_x	21.49	242.38	298.60	26.28	19.78	251.91	315.03	21.29
5 F_z	13.77	7.43	-26.40	10.61	16.49	-1.49	-36.82	13.14
6 F_x	76.90	68.82	245.02	15.36	17.30	197.57	129.34	101.20
6 F_z	2.37	11.95	-16.54	11.72	14.32	-16.60	-17.55	7.06
7 F_x	30.99	134.64	90.89	119.67	31.69	63.39	20.24	114.91
7 F_z	10.63	-11.15	5.19	-4.05	7.42	-7.84	8.26	-11.74
8 F_x	29.99	112.30	27.31	37.41	93.70	117.12	19.86	119.86
8 F_z	13.87	-10.68	13.34	9.01	-4.54	-17.91	10.88	-17.24
9 F_x	43.32	78.79	68.24	21.31	66.92	108.11	19.48	125.50
9 F_z	11.47	-4.42	7.80	10.23	-1.06	-18.03	9.69	-18.95
10 F_x	103.16	170.17	57.39	269.24	21.03	13.32	42.69	51.20
10 F_z	-2.67	-5.56	9.40	-36.83	11.86	10.12	1.69	8.73
11 F_x	28.80	99.44	33.76	24.86	90.54	109.52	22.93	114.64
11 F_z	13.35	-8.23	12.24	10.64	-4.98	-18.65	10.11	-17.74
12 F_x	37.36	48.75	270.98	94.64	107.81	16.26	11.05	384.16
12 F_z	15.91	6.93	-7.06	-2.23	-0.10	11.65	12.15	-40.52
13 F_x	23.40	95.81	23.16	54.19	112.26	76.11	74.82	75.04
13 F_z	12.62	-6.67	12.17	6.41	-8.11	-11.21	1.71	-10.19
14 F_x	85.67	52.69	26.80	44.85	89.57	98.61	91.56	42.53
14 F_z	3.87	-0.39	9.28	2.15	-1.29	-12.44	-2.41	-2.02
15 F_x	70.67	21.20	442.79	124.43	14.12	26.91	145.75	307.32
15 F_z	5.97	14.92	-25.45	-2.25	14.31	9.76	-19.27	-1.26
16 F_x	96.61	27.71	316.92	20.56	15.64	175.25	131.51	167.57
16 F_z	0.35	14.57	-18.38	11.45	14.25	-12.87	-17.14	4.51
17 F_x	227.68	18.81	22.23	124.03	52.72	123.33	67.08	30.85
17 F_z	-23.60	11.93	10.69	-9.49	10.80	-12.77	4.35	4.83
18 F_x	38.88	22.14	16.06	59.80	304.33	39.15	209.62	104.55
18 F_z	10.48	12.55	13.66	9.15	-28.04	3.34	-12.75	-11.67
19 F_x	142.07	19.07	208.49	104.84	101.23	68.68	11.11	333.35
19 F_z	-0.72	13.37	4.85	6.58	-3.75	-2.84	11.52	-32.30

Table 4 Cable tensions for the 19 solutions (Newton)

References

1. J.-P. Merlet, "The kinematics of the redundant N-1 wire driven parallel robot," in *IEEE Int. Conf. on Robotics and Automation*, Saint Paul, May, 14-18, 2012, pp. 2313–2318.
2. M. Carricato and J.-P. Merlet, "Stability analysis of underconstrained cable-driven parallel robots," *IEEE Trans. on Robotics*, vol. 29, no. 1, pp. 288–296, 2013.
3. M. Carricato and G. Abbasnejad, "Direct geometrico-static analysis of under-constrained cable-driven parallel robots with 4 cables," in *1st Int. Conf. on cable-driven parallel robots*, Stuttgart, September, 3-4, 2012, pp. 269–286.
4. Q. Jiang and V. Kumar, "The inverse kinematics of 3-d towing," in *ARK*, Piran, June 28- July 1, 2010, pp. 321–328.
5. J.-P. Merlet, "Managing the redundancy of N-1 wire-driven parallel robots," in *ARK*, Innsbruck, June, 25-28, 2012, pp. 405–412.
6. M. Such *et al.*, "An approach based on the catenary equation to deal with static analysis of three dimensional cable structures," *Engineering structures*, vol. 31, no. 9, pp. 2162–2170, 2009.
7. M. Gouttefarde *et al.*, "Simplified static analysis of large-dimension parallel cable-driven robots," in *IEEE Int. Conf. on Robotics and Automation*, Saint Paul, May, 14-18, 2012, pp. 2299–2305.
8. K. Kozak *et al.*, "Static analysis of cable-driven manipulators with non-negligible cable mass," *IEEE Trans. on Robotics*, vol. 22, no. 3, pp. 425–433, June 2006.
9. N. Riehl *et al.*, "Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, May, 14-16, 2009, pp. 2193–2198.
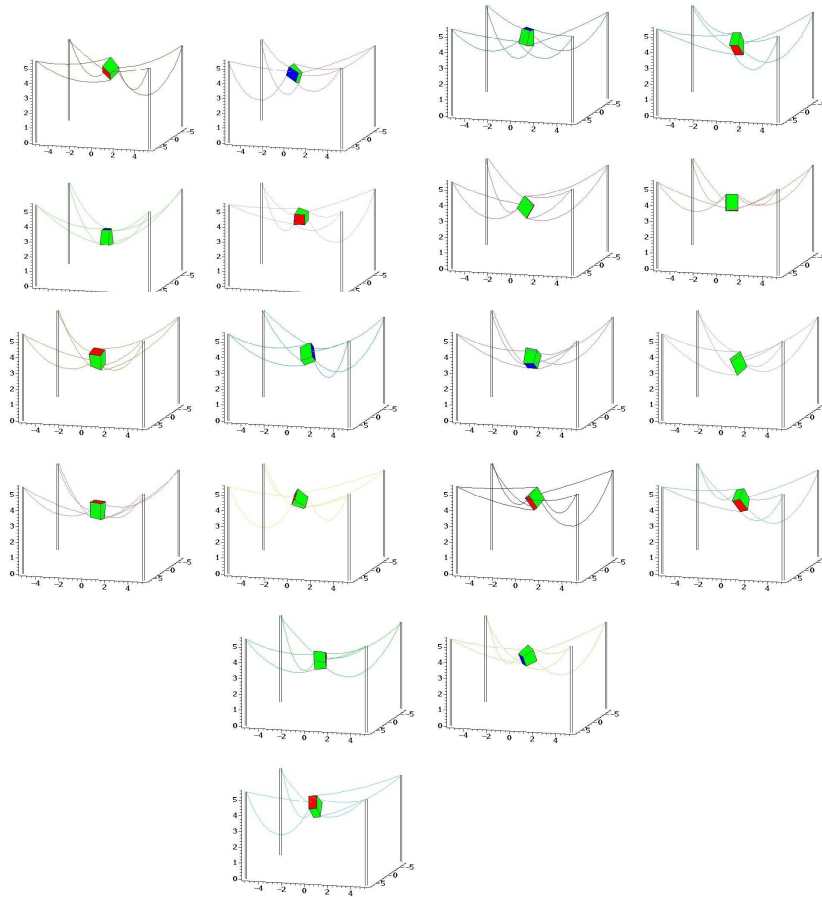10. H. M. Irvine, *Cable Structures*. MIT Press, 1981.

Fig. 4 Solutions 1 to 19

11. J.-P. Merlet, "Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis," *Int. J. of Robotics Research*, vol. 23, no. 3, pp. 221–236, 2004.