# Efficient estimation of the extremal articular forces of a parallel manipulator in a translation workspace

Jean-Pierre MERLET
INRIA Sophia-Antipolis
BP 93 06902 Sophia-Antipolis, France
E-mail: Jean-Pierre.Merlet@sophia.inria.fr

**Abstract**

*In this paper we consider a classical Gough platform with extensible legs whose platform is submitted to a given load. This load induces forces in the linear actuators of the legs, these forces being dependent upon the posture of the platform, and we want to determine the extremal values of the articular forces when the platform is translating in a given 3D workspace (the orientation of the platform is assumed to be constant). We describe an efficient algorithm which enable to compute the extremal forces more efficiently than a discretisation method.*

## 1 Introduction

In the design phase of a parallel manipulators it is extremely important to determine what will be the extremal articular forces of the platform induced by the presence of a given load on the moving platform. As an example we consider the classical Gough type parallel manipulator [4] illustrated in figure 1. In this robot a base and a platform are connected through 6 legs which have a ball-and-socket joint at each extremity. Linear actuators enable to change the leg lengths which in turn enable to control the position and orientation of the platform. If a given load is applied to the moving platform (for example the weight of some equipment) then each leg is submitted to a force acting along the leg axis. The values of these articular forces are position dependent and our purpose is to determine, for a given load, what will be the extremal values of the forces in the leg while the robot is moving in a given workspace. We will assume here that the workspace is only a translation workspace i.e. the orientation is kept constant and the workspace is defined by a 3D object which describe all the possible positions of the center $C$ of the moving platform. The

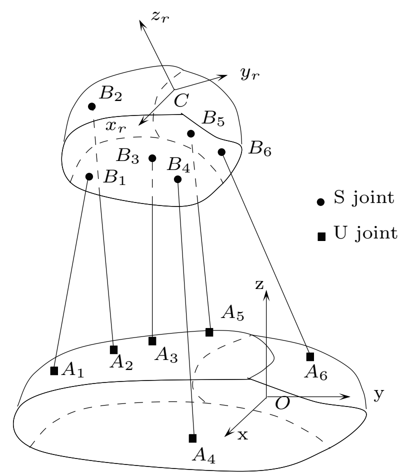


Figure 1: The classical Gough type parallel robot

coordinates of $C$ in the reference frame $(O, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ will be denoted $x, y, z$

Numerous papers have been devoted to the relations between articular forces and generalized forces acting on the platform [1, 2, 3, 5, 7, 8] but none, to the best of the author knowledge, have addressed our problem, which is however of practical importance. The computation time presented in this paper are established on a SUN Ultra 1 workstation.

## 2 Relation between the articular and generalized forces

Let $\boldsymbol{\mathcal{F}}$ denote the generalized forces applied on the moving platform and $\boldsymbol{\tau}$ the leg forces vector. It is well known that these quantities are related by:

$$\boldsymbol{\mathcal{F}} = J^{-T}\boldsymbol{\tau} \tag{1}$$

where $J^{-T}$ is the transpose of the inverse jacobian matrix of the robot, which is posture dependent. For the Gough platform a line $J_i$ of the inverse jacobian matrix may be written as:

$$J_i = (\frac{\boldsymbol{A_iB_i}}{||\boldsymbol{A_iB_i}||} \quad \boldsymbol{CB_i} \times \frac{\boldsymbol{A_iB_i}}{||\boldsymbol{A_iB_i}||}) \tag{2}$$

where $A_i(xa_i, ya_i, za_i), B_i(xb_i, yb_i, zb_i)$ are the extreme points of leg $i$. Note also that the leg length $\rho_i$ is equal to $||\boldsymbol{A_iB_i}||$. We define the matrix $H_i$ as the matrix obtained by substituting the i-th column of the matrix $J^{-T}$ by the vector $\boldsymbol{\mathcal{F}}$. Equation (1) defining a linear system we get each component $\tau_i$ of $\boldsymbol{\tau}$ by:

$$\tau_i = \frac{|H_i|}{|J^{-T}|} \tag{3}$$

Note that neither $|H_i|$ nor $|J^{-T}|$ are algebraic in terms of the coordinates of $C$ as $||\boldsymbol{A_iB_i}||$ appear in each matrix. A more convenient formulation will be presented now. Let the semi-inverse jacobian matrix $J_s^{-1}$ be defined by the 6 lines $J_s^i$:

$$J_s^i = (\boldsymbol{A_iB_i} \quad \boldsymbol{CB_i} \times \boldsymbol{A_iB_i})$$

and let $H_s^i$ be the matrix obtained by substituting the i-th column of $J_s^{-T}$ by the vector $\boldsymbol{\mathcal{F}}$. It is clear that:

$$|J^{-T}| = |J^{-1}| = \frac{|J_s^{-1}|}{\prod_{i=1}^{i=6} \rho_i} \tag{4}$$

By developing $|H_s^i|$ with respect to the i-th column we get:

$$|H_i| = \frac{|H_s^i|}{\prod_{j=1}^{j=6} \rho_j \quad j \neq i} \tag{5}$$

Note that both $|J_s^{-T}|, |H_s^i|$ are now algebraic in terms of the coordinates of $C$. Using these results and equation (3) we get:

$$\tau_i = \frac{\rho_i |H_s^i|}{|J_s^{-T}|} \tag{6}$$

Assume now that the coordinates of $C$ are functions of a parameter $r$. The derivative $D$ of $\tau_i$ with respect to $r$ can be computed as:

$$D = \frac{\partial \tau_i}{\partial r} = \frac{(\frac{\partial \rho^2}{\partial r}|H_s^i| + 2\rho^2 \frac{\partial |H_s^i|}{\partial r})|J_s^{-T}| - 2\rho^2 |H_s^i| \frac{\partial |J_s^{-T}|}{\partial r}}{2\rho |J_s^{-T}|^2} \tag{7}$$

Note that the numerator of this expression is algebraic in terms of $r$ and that the denominator is strictly positive. We assume here that there is no singularity in the workspace of the robot, this being verified using a method which will be described in another paper. In the sequel we will present a method to compute the extremum of the articular force for one leg, the process being identical for each leg.

# 3 Segment workspace

We assume that $C$ is moving on a given segment $M_1M_2$ and consequently we may write that:

$$\boldsymbol{OC} = \boldsymbol{OM_1} + \lambda \boldsymbol{M_1M_2}$$

where $\lambda$ is a scalar in the range [0,1]. To determine the extremal value of the articular force as $C$ moves on the segment it is sufficient to compute the roots $\lambda_i$ of the polynomial equation defined by the numerator $D_n$ of $D$ and then to compute the value of the articular force for each location of $C$ defined by the $\lambda_i$ included in the range [0,1] together with the force obtained for $\lambda = 0, 1$. Then the minimal and maximal values of these quantities are the extremal values of the articular force while $C$ is moving on the segment. Let us study in more details the degree of $D_n$: $|J_s^{-T}|$ is usually a third order polynomial in $\lambda$, $\rho^2$ is a second order polynomial in $\lambda$ while $|H_s^i|$ is a third order polynomial in $\lambda$. Consequently $D_n$ will be a seventh-order polynomial in $\lambda$.

We consider now a special case where the base is planar ($za_i = 0$) and the segment is oriented along the $x$ axis (or equivalently along the $y$ axis as we may rotate freely the reference frame around the $z$ axis) which will be useful in the sequel. In this case $D_n$ is of degree 6 as $|J_s^{-T}|$ is of degree 2 in $\lambda$. Furthermore if the platform is parallel to the base ($zb_i = \mathrm{C^{te}}$) then $|H_s^i|$ is of degree 1 in $\lambda$ and it may be shown that the degree of $D_n$ become 2 or 3 (this case will be denoted the *special case*).

The computation time of this procedure is about 55 ms for a general segment. If the base is planar and the segment is oriented along the $x$ axis the computation time is 15 ms. Although this computation time may seem to be relatively high most of it is devoted to the computation of some constant coefficients which are to be computed whatever the workspace is: consequently for more complex workspace for which determination of the articular force extremum on segments is needed the computation time will be deeply reduced.

# 4 Horizontal rectangle workspace

Now we assume that the workspace for $C$ is an horizontal rectangle defined by:

$$x_1 \leq x \leq x_2 \qquad y_1 \leq y \leq y_2$$

Note that vertical rectangle workspace can be treated in the same manner with an appropriate change in the direction of the reference frame.

### 4.1 General case

A first approach to determine the extremal values of the articular force will be to define two auxiliary variables $\alpha, \beta$ such that $x = x_1 + (1+\sin\alpha)(x_2-x_1)/2$, $y = y_1 + (1+\sin\beta)(y_2-y_1)/2$. Equation (7) will then be used to obtain two constraint equations in the unknowns $\sin\alpha, \sin\beta$. Unfortunately the degree of these equations is high (7 or 8) and it is difficult to determine the solution of this system.

We use therefore another approach which is to compute the articular force with an accuracy at least better than a given constant $\epsilon$. The idea is first to compute the articular force on the segment $x_1 \leq x \leq x_2, y = y_1$ using the result of the previous section. Let $\tau_{min}, \tau_{max}$ be the force computed at this stage. We then investigate the minimal value of a variable $y^2_\Delta$ such that on the segment $x_1 \leq x \leq x_2, y = y_1 + y^2_\Delta$ the corresponding articular force is equal to $\tau_{min} - \epsilon$ or $\tau_{max} + \epsilon$. In other words if $\tau^s_{min}(y), \tau^s_{max}(y)$ denote the minimal and maximal articular force on the segment $x_1 \leq x \leq x_2, y$ we have to find the minimal value of $y^2_\Delta$ such that

$$\begin{aligned} &\tau^s_{min}(y_1 + y^2_\Delta) = \tau_{min} - \epsilon \quad \text{or} \\ &\tau^s_{max}(y_1 + y^2_\Delta) = \tau_{max} + \epsilon \end{aligned} \tag{8}$$

If we define $x = x_1 + \lambda(x_2 - x_1)$ these equations are functions of $y^2_\Delta, \lambda$. Here the value of $y^2_\Delta$ is assumed to be small so that equations (8) may be developed at first order to become:

$$A_2(\lambda) y^2_\Delta + A_0(\lambda) = 0$$

Thus the value of $y^2_\Delta$ is obtained as a function of $\lambda$ by:

$$y^2_\Delta = -\frac{A_0(\lambda)}{A_2(\lambda)}$$

The derivative of $y^2_\Delta$ with respect to $\lambda$ is computed and lead to a 22nd order polynomial in $\lambda$. The value of $y^2_\Delta$ is computed for each root of this polynomial in the range [0,1] together with the value at $\lambda = 0, 1$ and the minimal value of $y^2_\Delta$ is retained. If this value is small the computation is supposed to be exact otherwise a fixed small value (0.3 in our current implementation) is assigned to $y^2_\Delta$. The extremal articular force are computed on the segment $x_1 \leq x \leq x_2, y = y_1 + y^2_\Delta$ and the values of $\tau_{min}, \tau_{max}$ are updated. This process is repeated until the value of $y$ is equal to $y_2$.

In order to speed up this analysis we have computed the derivative of $\tau$ as function of $y$. Remember that the denominator of this derivative has a constant sign and that the numerator is algebraic in terms of $x, y$. As $x, y$ are bounded a simple interval analysis enable to estimate what will be the minimum and maximum of this derivative. If they are found to be of constant sign then $\tau$ is monotonous with respect to $y$ and the extremum of the articular force are obtained by computed the articular force on the two segments $x_1 \leq x \leq x_2, y = y_1$, $x_1 \leq x \leq x_2, y = y_2$.

### 4.2 Special case

As mentioned previously if both the base and the platform are planar we get a simplification in $D_n$ which enable to use the optimization approach. In that case we may find the extremum of the articular forces by solving two pairs of second order equations.

For articular forces in the range of 200 and an accuracy of 1 the computation time is about 2500 ms for a general robot and 1600 ms if the base is planar. In the special case this time is reduced to 40 ms.

## 5 Box workspace

### 5.1 Principle

Now we assume that the workspace for $C$ is a box defined by:

$$x_1 \leq x \leq x_2 \qquad y_1 \leq y \leq y_2 \quad z_1 \leq z \leq z_2$$

In view of the previous section it is clear that the optimization approach cannot be used. Therefore we use a similar approach as for the rectangle workspace. We first compute the extremum articular force in the rectangle $x_1 \leq x \leq x_2, z = z_1$ using the result of the previous section. Let $\tau_{min}, \tau_{max}$ be the current extremal articular forces. We then investigate the minimal value of a variable $z^2_\Delta$ such that on the rectangle $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z = z_1 + z^2_\Delta$ the corresponding articular force is equal to $\tau_{min} - \epsilon$ or $\tau_{max} + \epsilon$. In other words if $\tau^s_{min}(z), \tau^s_{max}(z)$ denote the minimal and maximal articular force on the rectangle $x_1 \leq x \leq x_2, y_1 \leq y \leq y_2, z$ we have to find the minimal value of $z^2_\Delta$ such that:

$$\begin{aligned} &\tau^s_{min}(z_1 + z^2_\Delta) = \tau_{min} - \epsilon \quad \text{or} \\ &\tau^s_{max}(z_1 + z^2_\Delta) = \tau_{max} + \epsilon \end{aligned} \tag{9}$$

If we define $x = x_1 + \lambda(x_2 - x_1)$, $y = y_1 + \mu(y_2 - y_1)$ these equations are functions of $z^2_\Delta, \lambda, \mu$. Here the value of $z^2_\Delta$ is assumed to be small so that equations (9) may be developed at first order to become:

$$A_2(\lambda, \mu) z^2_\Delta + A_0(\lambda, \mu) = 0$$

the value of $z_\Delta^2$ is obtained as a function of $\lambda, \mu$ by:

$$z_\Delta^2 = -\frac{A_0(\lambda, \mu)}{A_2(\lambda, \mu)}$$

The minimal value of $z_\Delta^2$ if obtained for $\lambda, \mu$ such that the derivatives of $z_\Delta^2$ with respect to $\lambda, \mu$ vanish. As $\lambda, \mu$ belong to [0,1] these derivatives are approximated to the second order around the value 0, 0.5, 1. Hence for each case we get two second order equations in these variables and their resultant is a fourth order polynomial in $\lambda$. By solving this polynomial we get all the possible values of $\lambda$ and for each $\lambda$ we get corresponding values of $\mu$. For each pair $(\lambda, \mu)$ we then compute the value of $z_\Delta^2$ and we retain the smallest positive value. The value of $z$ is updated to $z + z_\Delta^2$ if the computed value of $z_\Delta^2$ is sufficiently small otherwise a small value is assigned to $z_\Delta^2$. This process is repeated until the value of $z$ is equal to $z_2$.

### 5.2 Speeding up the algorithm

In order to speed up this analysis we have computed the value of $\tau$ as function of $x, y, z$. Remember that $\tau_i$ may be written as $\rho_i |H_s^i| / |J_s^{-T}|$ where both $|H_s^i|, |J_s^{-T}|$ are algebraic in terms of $x, y$. As $x, y$ are bounded a simple interval analysis enable to estimate what will be the minimum and maximum of $|H_s^i| / |J_s^{-T}|$. Another trivial algorithm enable to compute the maximal and minimal value of $\rho_i$ in the rectangle. Consequently it is easy to find bounds on the value of $\tau_i$ in the rectangle at altitude $z$. If these bounds lie within the current range $[\tau_{min}, \tau_{max}]$, then we skip the computation of the current rectangle and compute a new value of $z_\Delta^2$.

Similarly we compute the derivative of $\tau$ with respect to $z$: we get therefore an expression function of $x, y, z$. A simple analysis interval enable to compute the extremum values of this derivative and if the minimum and maximum are of same sign then $\tau$ is monotonous with respect to $z$ and the extremum of the articular forces are obtained by computing the extremum for the rectangles at altitude $z_1, z_2$.

### 5.3 Computation time

For articular forces in the range of 200 and an accuracy of 1 the computation time vary from 6400 to 21000 ms for a general robot and from 3000 to 4000 ms if the base is planar (the difference between the two types of robot are mainly due to a more careful implementation of the case of the planar base). In the later case as the computation of the articular forces for one posture of the robot take about 0.5 ms a discrete method will have split each three main axis in 18 to 20 points (hence we may miss postures where important articular forces occur).

The sensitivity of the computation time to the accuracy with which the extremum of the articular forces are computed is presented in figure 2. It may be seen

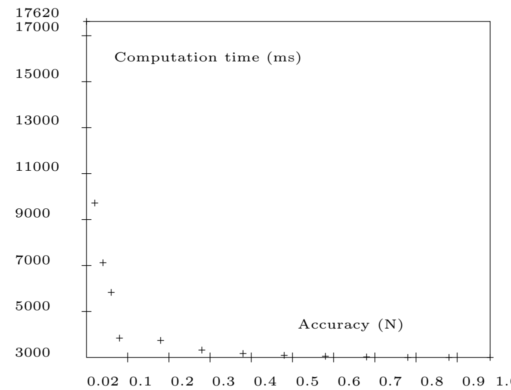


Figure 2: Computation time versus the desired accuracy for a box workspace and a robot with a planar base

that the computation time is prohibitive only for an accuracy which is far away from the usual one necessary for the design process.

### 5.4 Special case

In the special case the optimization approach can be used. The constraint equations can be reduced to solving a set of 4 univariate polynomials of degree 2,3,4,8. This enable to get the exact extremum and reduce the computation time to 120 to 200 ms.

## 6 Polyhedric workspace

The low computation time for the determination of the extremal articular forces when the workspace is a box suggests that this method can be extended to more complex workspaces. We will assume here that a workspace is described by a set of polygonal cross-sections. We will compute the extremal articular forces for each volume defined between two successive cross-sections, then it will be easy to determine the extremal articular forces for the whole workspace. A given volume will be decomposed into as many boxes as necessary until the articular forces are determined with the desired accuracy (note that it is trivial to determine if a box lie within the workspace). A list of box $\mathcal{B}$ is maintained during the algorithm: this list is initialized with the bounding box $B_0$ of the whole volume. A set of extremum articular forces is initialized by computing the articular forces at some vertex of the workspace. The range of these forces will be called the current articular forces range. At step k the algorithm perform the following operations:

1. if the box $B_k$ is completely outside the volume we consider the next box in the list
2. if the box $B_k$ lie completely within the workspace we compute the extremal articular forces for this box and update the current articular forces range
3. if the box $B_k$ lie partially within the workspace we compute the extremal articular forces for this box
   (a) if these forces lie within the current articular forces range we consider the next box in the list
   (b) otherwise the box is split into eight boxes by dividing each dimension of the box by 2. The resulting boxes are put at the end of the list and we consider the next box in the list.

The algorithm stop if there is no more box in the list. Note that to speed up the process an heuristic is used: whenever the extremum for a box has to be computed we first estimate a bound on the value of the extremum by using an interval analysis similar to the analysis presented in the section devoted to the box workspace: if these bounds lie within the current articular forces range we skip the computation for this box and moves to the next box in the list.

The computation time is reasonable: for example we have considered the workspace defined by three square cross-sections: $z = 50, x \in [-10, 10], y \in [-10, 10]$, $z = 55, x \in [-5, 5], y \in [-5, 5]$, $z = 60, x \in [-10, 10], y \in [-10, 10]$ represented in figure 3. For a

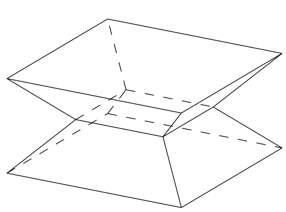

Figure 3: The test volume

general robot the computation time vary from to 65s to 165s while when the base is planar the computation time vary from 7s to 9s (1.2s in the special case). The sensitivity of the computation time with respect to the accuracy is illustrated on figure 4. It may be seen that

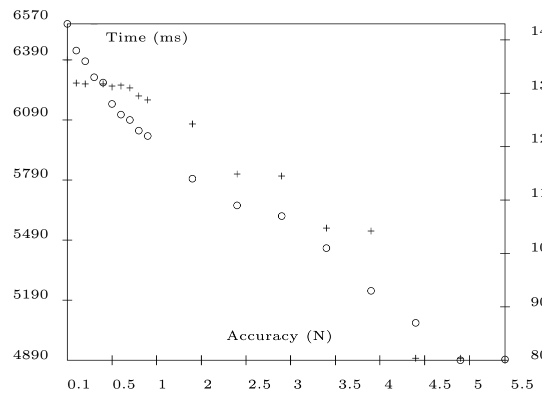


Figure 4: Computation time versus the desired accuracy for the test volume for a robot with a planar base (the crosses represent the general case while the circles represent the special case) .

even for an accuracy of 0.1 N the computation time is reasonably low at 6570 ms.

## 7 Articular workspace

Let us assume that the leg length have a minimal and a maximal values $\rho_{min}, \rho_{max}$. It may be of interest to compute the extremum of the articular forces in the workspace defined by a constant orientation and any position of the platform which fulfill the constraints on the leg lengths: this workspace will be called the *articular workspace*. A simple adaptation of the previous algorithm enable to perform this task. Note first that a trivial algorithm enable to determine what will the extremum of each leg lengths while $C$ moves in a given box: we will denote this algorithm $Max_\rho(B)$ where $B$ is a box. Then notice that it is easy to determine a box which contain all the possible locations of the platform being given the extremal values of the leg lengths. We start the previous algorithm with this box. Then we have to change the inclusion test in the previous algorithm: a box will lie within the workspace if all the ranges given by $Max_\rho(B)$ lie within $[\rho_{min}, \rho_{max}]$ while a box will be completely outside the workspace if one of the ranges is outside the range $[\rho_{min}, \rho_{max}]$. In any other cases we assume that the box is partially within the workspace. Strictly speaking this may be false: as $Max_{\rho}(B)$ gives the extremum of the leg lengths *independently* it may occur that there is no posture of the platform where *all* the leg lengths lie in the correct range at the same time, but as this type of box will be divided in smaller box during the process our assumption lie on the safe side.

In the implementation of this algorithm the following precautions have to be taken:

- if the leg lengths for a given box exceed by far the articular limits while still the box is partially inside the articular workspace it is better to split the box without computing the extremum of the articular forces for this box (to avoid computation for large boxes)
- as soon as all the dimensions of a box are quite small we consider that the articular forces in the box are given for the forces obtained at the center of the box. This avoid to create a large number of boxes in the case where the center of the box is on the boundary of the workspace

With these heuristics the computation time is reasonable: typically 4.5 mn for the full articular workspace for an accuracy of 1 (approximatively 2.5 mn in the special case) and 100 s for an accuracy of 5 (60s in the special case) but we still have the guarantee on the validity of the result. Note also that another approach will be to use the algorithm described in [6] which enable to compute exact cross-sections of the workspace for a constant orientation and then use the algorithm described for the polyhedric workspace.

## 8 Conclusion

The algorithm presented in this paper enable to compute efficiently one of the most important feature for the design of a parallel robot. Although the computation time may seem to be high it must be noted that for each type of workspace we have find numerous examples for which the computation time of a discretisation method necessary to determine the articular forces with the same level of accuracy exceed by far the computation time of our algorithm. It has also been noted that the discrepancy between the results of our algorithm and of a discretisation method with a similar computation time may reach up to 10 %.

Still the workspace we have been considering is only the translation workspace but a discretisation on the 3D orientation workspace will be by far less computer expensive than the discretisation on the full 6D workspace. This algorithm will be integrated in the near future in our design methodology `DEMOCRAT` for the design of parallel robot.

Note that this algorithm can also be extended to other mechanical architecture of fully-parallel 6 DOF robots as most of them have an inverse jacobian matrix similar to the matrix of the Gough platform.

## References

[1] Agrawal S.K. and Roth B. Statics of in-parallel manipulator systems. *ASME J. of Mechanical Design*, 114:564–568, December 1992.

[2] Bryfogle M.D., Nguyen C.C., Zhou Z-l., and Antrazi S.S. A methodology for geometry design of closed kinematic chain mechanisms. In *IEEE Int. Conf. on Robotics and Automation*, pages 2974–2979, Albuquerque, April, 21-28, 1997.

[3] Duffy J. *Statics and Kinematics with Applications to Robotics*. Cambridge University Press, New-York, 1996.

[4] Gough V.E. and Whitehall S.G. Universal tire test machine. In *Proceedings 9th Int. Technical Congress F.I.S.I.T.A.*, volume 117, pages 117–135, May 1962.

[5] Kosuge K. and others . Input/output force analysis of parallel link manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 714–719, Atlanta, May, 2-6, 1993.

[6] Merlet J-P. Détermination de l'espace de travail d'un robot parallèle pour une orientation constante. *Mechanism and Machine Theory*, 29(8):1099–1113, November 1994.

[7] Orin D.E. and Oh S.Y. Control of force distribution in robotic mechanisms containing closed kinematic chains. *J. of Dyn. Syst. Meas. and Control*, 102:134–141, June 1981.

[8] Pang H. and Shahinpoor M. Analysis of static equilibrium of a parallel manipulator. *Robotica*, 11:433–443, 1993.