

A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators

Lars Mikelsons, Tobias Bruckmann, Manfred Hiller, Dieter Schramm

Angaben zur Veröffentlichung / Publication details:

Mikelsons, Lars, Tobias Bruckmann, Manfred Hiller, and Dieter Schramm. 2008. "A real-time capable force calculation algorithm for redundant tendon-based parallel manipulators." In *2008 IEEE International Conference on Robotics and Automation, 19-23 May 2008, Pasadena, CA, USA*, edited by Seth Hutchinson, 3869–74. Piscataway, NJ: IEEE.
<https://doi.org/10.1109/robot.2008.4543805>.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:

Deutsches Urheberrecht

Weitere Informationen finden Sie unter: / For more information see:

<https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/>



A Real-Time capable Force Calculation Algorithm for redundant Tendon-Based Parallel Manipulators

Lars Mikelsons, Tobias Bruckmann, Manfred Hiller, Dieter Schramm

Abstract—Tendon-based parallel manipulators with n d.o.f. use at least $m = n + 1$ tendons to guide the endeffector along a given trajectory. Since tendons can only transmit limited and tractive forces, it is essential to apply a valid tendon force distribution. Due to safety and precision requirements, a combined position and force control is needed where the force calculation delivers the desired tendon force distributions. The high dynamic potential of the robot demands for realtime capable algorithms. To avoid steps in the motor torques the calculated tension force distributions also have to be continuous along the trajectory. In this paper, a new algorithm for tendon force distribution calculations capable for usage on a realtime system is proposed and its continuity is proven.

I. INTRODUCTION

At the Chair of Mechatronics, a testbed for tendon-based Stewart-platforms (SEGESTA *Seilgetriebene Stewart-Plattformen in Theorie und Anwendung*) has been developed during the past few years. Presently, the SEGESTA teststand has $n = 6$ d.o.f. and uses $m = 7$ tendons to move the platform along desired trajectories [8]. Optionally an eighth tendon can be added. For low dynamic movements it is possible to guide the endeffector along a trajectory only with position control. In this case the inverse kinematics delivers the tendon length for the corresponding position. Sensors for actual tendon length are installed so feedback control is reasonable. However for high dynamic movements only position control is not sufficient since the endeffector begins to wobble. Since this is due to loose tendons, force control is needed for accurate and fast movement. Additionally based on the calculated force distribution workspace, boundary detection can be performed. If $m = n + 1$ tendons are used, the force distribution calculation is straight forward. The case of $m > n + 1$ tendons demands for more sophisticated methods. Here optimization methods are widely used ([15],[12], [3], [16], [2]). Standard optimizer implementations ([1], [11]) require iterative computations which may not be suitable within a realtime controller system due to their usually non-predictable worst-case runtime. [13] transforms the problem to a linear programming problem which allows a direct calculation of the minimum force distribution. However this does not guarantee continuity. Alternatively quadratic programming which is usually using iterative methods is used to get continuous solutions. Here again the problem of non-predictable worst-case runtime occurs.

This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under HI370/24-1 and SCHR1176/1-2

The authors are with the Chair for Mechatronics, University of Duisburg-Essen, 47057 Duisburg, Germany [bruckmann, mikelsons, hiller,schramm@imech.de](mailto:bruckmann@mikelsons.hiller,schramm@imech.de)

The paper is structured as follows: In section II, a short description of the kinematics and dynamics of SEGESTA is given. Methods for generating continuous tendon force distributions are discussed in section III, including an example. The proof of continuity is given in section IV. Finally, in section V, the conclusions are drawn and an outlook is given.

II. KINEMATICS AND FORCE EQUILIBRIUM

SEGESTA consists of two main components: a frame of aluminium profile bars which carries motors, winches as well as further components like computers, measurement equipment etc. (fig. 1). The triangular shaped platform is connected to the winches by tendons. SEGESTA is designed as an reconfigurable system by using modules which carry winches and motors and which can be installed and removed easily. Due to its lightweight structure, SEGESTA can generate high-dynamic motions [7].

Following a trajectory, at every time step the endeffector pose is calculated. The corresponding tendon length is delivered by the inverse kinematic which is easy to evaluate. A general tendon-based parallel manipulator can be described using the following vectors and coordinate frames, with $\mu = 1, \dots, m$ [5]:

- The coordinate frame C_B is the base frame, while C_P is connected to the platform (fig. 2).
- The vectors b_μ denote the positions of the winch points, represented by the points where tendons are led through small ceramic eyes which are fixed.
- p_μ are the platform-fixed vectors to the connecting points
- l_μ denote the tendon vectors from the platform to the winches.
- The forces in the tendons are described by f_μ , where f_P and τ_P denote all other applied forces and torques acting on the platform.

Due to the fact that tendons can only transmit tractive forces the tendons always have to be tensed. Thus at least $n + 1$ tendons are needed to tense the system, if no external loads are present [9]. Setting up the force equilibrium for the endeffector leads to ([10],[15])

$$\begin{bmatrix} \nu_1 & \dots & \nu_m \\ p_1 \times \nu_1 & \dots & p_m \times \nu_m \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} + \begin{bmatrix} f_P \\ \tau_P \end{bmatrix} = 0 \quad (1)$$

with $\nu = \frac{l_\mu}{|l_\mu|}$ and $f > 0$ or in a more compact form as

$$A^T f + w = 0, \quad f > 0. \quad (2)$$



Fig. 1. SEGESTA Testbed

III. SAFE FORCE GENERATION METHOD

Since force control is necessary to guarantee a defined tension distribution, a method to calculate tendon forces must be provided. Because of the force redundancy $r = m - n$ in the considered systems ($m \geq n + 1$), there exists a r dimensional solution set for the force distribution. In the case of an one-dimensional solution set, the calculation of the force distribution is clearly straight-forward. In the case of higher redundancies, more advanced techniques are needed. Typically a cost function and linear constraints on the forces are used to ensure a continuous solution along a trajectory. Usually an optimizer uses iterative steps which leads to non-predictable worst-case runtime. Hence those methods are not applicable within a realtime controller. This justifies the need of a realtime capable force distribution calculation algorithm. In this paper, a non-iterative algorithm which provides continuous force distributions furthestmost from the force limits is proposed and a proof for its continuity given. The algorithm provides a force distribution which leads to a fairly tensed system. Due to this property, the obtained solution is called a safe solution. Beside minimum tendon forces (which can be zero as smallest possible force) also the maximum tendon forces are of great importance since their ratio defines the workspace boundaries. To evaluate the proximity of a specific position of the platform to the workspace boundaries, knowledge of the tendon forces is presumed. Obtaining a solution from the optimization algorithm which exceeds the tendon force boundaries means that the platform is outside the predefined workspace. So, the calculation of force distributions plays also an important role in terms of reliability and safety. In practice, it is of great importance to find continuous solutions. Non-continuous tendon forces

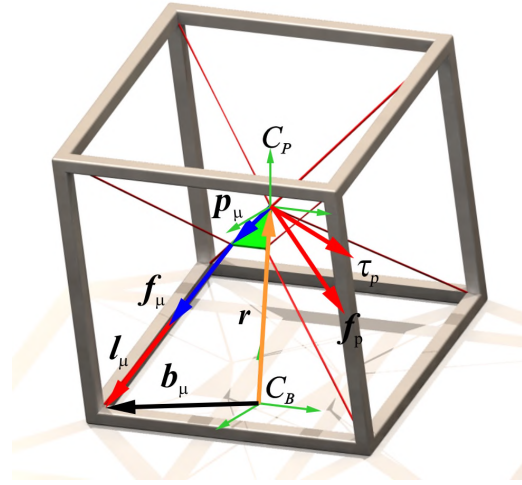


Fig. 2. Symbol Definitions for a General Tendon-Based Stewart-Platform

may consist of acceptable solutions, but since those values are needed for control, they would cause steps in motor torques which leads to vibrations and high mechanical loads. Using both the results from the inverse kinematics and the force optimization makes a combined position-force-control possible. The position part delivers positioning precision while the force controller is responsible for positive tensions and acts as a kind of pilot control. Detailed concepts for tendon force control are also proposed in [13].

A. Safe Force Calculation

In this section the force calculation algorithm is presented. For an arbitrary tendon-based parallel manipulator, the structure matrix $A^T \in \mathbb{R}^{n \times m}$ can be easily obtained for every pose. To get solutions for the force equilibrium in eqn. 1, the kernel spanned by r base vectors h_1, \dots, h_r has to be considered. For the moment, the homogenous case of $w = 0$ is considered which will be extended later. Since m tendons are present, the force limits form an m -dimensional hypercube $C \subset \mathbb{R}^m$. All force distributions satisfying the force equilibrium in eqn. 1 form a r -dimensional subspace $S \subset \mathbb{R}^m$ spanned by the kernel of the structure matrix. If the intersection F of the hypercube C and the subspace S is non-empty, solutions f in the acceptable solution set F exist, i.e. $F = C \cap S \neq \emptyset$, where F is a r -dimensional manifold of the \mathbb{R}^m . The required calculation of the kernel can be done effectively using a QR decomposition of A^T . The matrix $Q \in \mathbb{R}^{n \times n}$ is orthogonal which implies that only the trapezoidal $R \in \mathbb{R}^{n \times m}$ has to be considered to get the kernel of A^T . The kernel is used to define a map H from the \mathbb{R}^r to $S \subset \mathbb{R}^m$ (see fig. 3) by

$$H = [H_1 \quad \dots \quad H_r] \in \mathbb{R}^{m \times r}. \quad (3)$$

Then for all $\lambda \in \Lambda$, the following must hold, where Λ is the (convex) polyhedron-shaped preimage of the manifold F

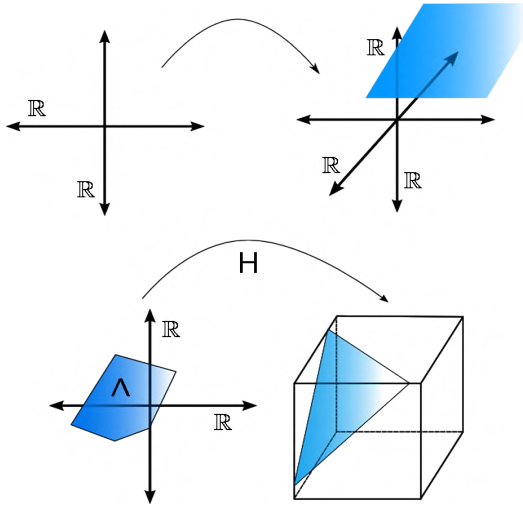


Fig. 3. Usage of the kernel as a map in the case of $m=3$ and $n=1$

under the mapping H :

$$f_{\min} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^{m \times 1} \leq H\lambda \leq f_{\max} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^{m \times 1} \quad (4)$$

In other words, since the kernel H maps the \mathbb{R}^r onto the solution subspace S , it maps the polyhedron $\Lambda \subset \mathbb{R}^r$ onto the solution manifold $F \subset \mathbb{R}^m$. To get a more convenient characterization of Λ , in the next step the vertices of Λ are calculated. Component-wise evaluation of ineqn. 4 for both sides of the inequalities gives $2m$ hyperplanes in \mathbb{R}^r . The vertices of the polyhedron are intersection points of at least r of those hyperplanes. Hence all intersection points are calculated by considering all possible combinations of $r \times r$ systems of linear equations in λ . Every solution is examined with respect to its compliance with all inequalities of eqn. 4. If a solution satisfies all inequalities, it is a vertex of the convex polyhedron Λ . Clearly, a vertex may be the solution of more than one linear equation system, satisfying all inequalities. Nevertheless, they describe the same vertex and thus, the solution is only considered once for the next steps. With the knowledge of the vertices, Λ can now be triangulated, i.e. Λ is splitted into r -simplexes. In the case of $r = 2$ this just means dividing into triangles. In higher dimensions advanced techniques as presented in [4] are required. Triangulation delivers a list of n_s simplexes P^k with each having $r + 1$ vertices v_{k_j} with $k = 1 \dots n_s$ and $j = 1 \dots r + 1$. After triangulation, the volumes V^k of the simplexes are determined by integration [6]. Furthermore their CoG λ_{s_k} is computed by

$$\lambda_{s_k}^i = \frac{\sum_{\nu=1}^{r+1} v_{k_\nu}^i}{r+1} \quad i = 1 \dots r \quad k = 1 \dots n_s \quad (5)$$

which is used to calculate the CoG λ_s of the polyhedron via

$$\lambda_s^i = \frac{\sum_{\mu=1}^{n_s} (\lambda_{s_\mu}^i \cdot V^\mu)}{\sum_{\mu=1}^{n_s} V^\mu}. \quad (6)$$

Finally, the solution is transformed back using the kernel H as a map

$$x_s = H\lambda_s \quad (7)$$

where x_s is the center of the manifold F .

In the inhomogenous case, a non-zero load w is to be considered.

$$A^T f + w = 0, \quad f > 0. \quad (8)$$

Thus, the solution subspace S has the form

$$f = p + H\lambda, \quad \lambda \in \mathbb{R}^r, \quad (9)$$

where p denotes a particular solution of eqn. 8. Obviously external loads shift the solution plane. The particular solution is computed by

$$A^T p + w = 0 \Leftrightarrow Q R p + w = 0 \quad (10)$$

First, a intermediate solution y is obtained by

$$Q y = -w \Leftrightarrow y = -Q^T w \quad (11)$$

Finally, p is computed by concerning the underdetermined system

$$R p = y \quad (12)$$

and freely choosing r parameters.

Instead of moving the plane, the cube of the force limits is moved by $-p$ which is done by subtracting p on both sides of eqn. 4 before performing the algorithm. The final result for the CoG must be transformed back by adding p .

B. Proof-of-Concept

In this section it is shown that the CoG of the manifold F can be computed by calculating the CoG of the convex polyhedron. The CoG of a general body can be computed componentwise as

$$x_s^i = \frac{\int_F x^i dF}{V(F)}. \quad (13)$$

Now, the theorem for integration on manifolds states

$$x_s^i = \frac{\int_{\Lambda} x^i \circ H^* \sqrt{\det((DH)^*{}^T (DH)^*)} d\lambda}{\int_{\Lambda} 1 \circ H^* \sqrt{\det((DH)^*{}^T (DH)^*)} d\lambda} \quad (14)$$

where $H^* : \Lambda \mapsto F$, $\lambda \mapsto H\lambda$ is a linear map from Λ to F and $(DH)^*$ is the Jacobian of H^* which is equal to H itself

$$(DH)^* = \frac{\partial H^*}{\partial \lambda} = H.$$

Furthermore, $\sqrt{\det(\mathbf{H}^T \mathbf{H})}$ is independent from λ and can therefore be canceled in the next step. Additionally splitting Λ into the simplexes gives:

$$x_s^i = \frac{\sum_{\nu=1}^{n_s} \int_{P^\nu} x^i \circ \mathbf{H}^* d\lambda}{\sum_{\nu=1}^{n_s} \int_{P^\nu} 1 d\lambda} \quad (15)$$

Now the expression $x^i \circ \mathbf{H}^* = \sum_{\mu=1}^r \mathbf{H}_{\mu,i} \lambda^\mu$ can be inserted which results in

$$x_s^i = \frac{\sum_{\nu=1}^{n_s} \int_{P^\nu} \sum_{\mu=1}^r \mathbf{H}_{\mu,i} \lambda^\mu d\lambda}{\sum_{\nu=1}^{n_s} V^\nu} \quad (16)$$

Since \mathbf{H} is independent from λ , it can be moved out of the integral. In vector form

$$x_s^i = \frac{[H_{1,i} \ \dots \ H_{r,i}]}{\sum_{\nu=1}^{n_s} V^\nu} \cdot \begin{bmatrix} \sum_{\nu=1}^{n_s} \int_{P^\nu} \lambda^1 d\lambda \\ \vdots \\ \sum_{\nu=1}^{n_s} \int_{P^\nu} \lambda^r d\lambda \end{bmatrix}$$

Because of eqn. 13, this equation is rewritten:

$$x_s^i = \frac{[H_{1,i} \ \dots \ H_{r,i}]}{\sum_{\nu=1}^{n_s} V^\nu} \begin{bmatrix} \sum_{\nu=1}^{n_s} \lambda_{s_\nu}^1 \cdot V^\nu \\ \vdots \\ \sum_{\nu=1}^{n_s} \lambda_{s_\nu}^r \cdot V^\nu \end{bmatrix} \quad (17)$$

Using eqn. 6, one gets

$$x_s^i = [H_{1,i} \ \dots \ H_{r,i}] \begin{bmatrix} \lambda_s^1 \\ \vdots \\ \lambda_s^r \end{bmatrix} = [H_{1,i} \ \dots \ H_{r,i}] \lambda_s$$

Therefore $x_s = \mathbf{H} \lambda_s$ holds where λ_s denotes the CoG of Λ in \mathbb{R}^r .

C. Example

In this section the algorithm will be applied to a planar manipulator having 2 translational d.o.f. ($n = 2$) neglecting external wrenches. It is driven by 4 tendons ($m = 4$). Thus its structure matrix ($A^T \in \mathbb{R}^{2 \times 4}$) has a 2 dimensional kernel ($r = m - n = 2$). Though this is a very simple robot, it is well designed due to the relation between n and m ($m = 2n$) [14]. According to fig. 4 one gets for the shown position the structure matrix

$$A^T = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

The base vectors of the kernel of A^T can be easily calculated as

$$h_1 = [1 \ 0 \ 1 \ 0]^T \quad h_2 = [0 \ 1 \ 0 \ 1]^T.$$

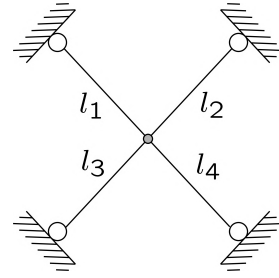


Fig. 4. Planar robot with 2 d.o.f. and 4 tendons (base length 1m)

The linear mapping

$$\mathbf{H} := [\mathbf{h}_1 \ \mathbf{h}_2]$$

maps the \mathbb{R}^2 onto the 2-dimensional plane of valid force distributions in \mathbb{R}^4 , i.e. force distributions which satisfy the force equilibrium. The set of force distributions which lie inside the force limits forms a hypercube in \mathbb{R}^4 . Hence the intersection of the hypercube and the plane is the set of acceptable force distributions. Consequently, the set of acceptable solutions can be described by a convex set $\Lambda \subset \mathbb{R}^2$ as follows

$$\Lambda := \left\{ \lambda \in \mathbb{R}^2 \mid f_{\min} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \leq \mathbf{H} \lambda \leq f_{\max} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}, \quad (18)$$

i.e. Λ is the preimage of the acceptable solution set under \mathbf{H} . Here f_{\min} and f_{\max} denote the lower and the upper force limit, respectively. It is obvious that Λ is convex and bounded by lines, i.e. a polyhedron. Thus it is possible to characterize Λ by its vertices. The intersections of the lines which bound Λ have to be calculated. These lines are given by componentwise examination of the two inequalities in 18. Setting f_{\min} to $1N$ and f_{\max} to $100N$ the four lines

$$1 = \lambda_1; \quad 100 = \lambda_1 \quad (19)$$

$$1 = \lambda_2; \quad 100 = \lambda_2 \quad (20)$$

bound the polyhedron. This leads to the intersection points $(1N, 1N)$, $(1N, 100N)$, $(100N, 1N)$, $(100N, 100N)$. Since all intersection points satisfy all inequalities of 18, every intersection point is a vertex of the polyhedron. Due to the convexity of the polyhedron triangulation can be easily performed by choosing one vertex and connecting all vertices to the initial one. The choice of $(1N, 1N)$ gives two triangles. T_1 with vertices $(1N, 1N)$, $(100N, 1N)$, $(100N, 100N)$ and T_2 with vertices $(1N, 1N)$, $(1N, 100N)$, $(100N, 100N)$. One can calculate the CoG s_1 of T_1 as $s_1 = (67N, 34N)$ and its volume V_1 as $V_1 = 4900,5N^2$. For T_2 one gets $s_2 = (34N, 67N)$ and $V_2 = 4900,5N^2$. This results in $s_P = (50.5N, 50.5N)$ for the CoG s_P of the polyhedron. Using the linear mapping \mathbf{H} the final result is the CoG $s = \mathbf{H} s_P = (50.5N, 50.5N, 50.5N, 50.5N)$ of the acceptable solution plane.

D. Results

In this example the endeffector of a tendon-based stewart platform has to follow a screwline with increasing radius (see fig. 5). The robot is designed as described in section I with frame dimensions $800 \times 2000 \times 1500 \text{ mm}^3$. Eight tendons are used. Thus the CoG of a two dimensional manifold in the \mathbb{R}^8 has to be calculated. The force limits are $10N$ for the lower bound and $1000N$ for the upper bound.

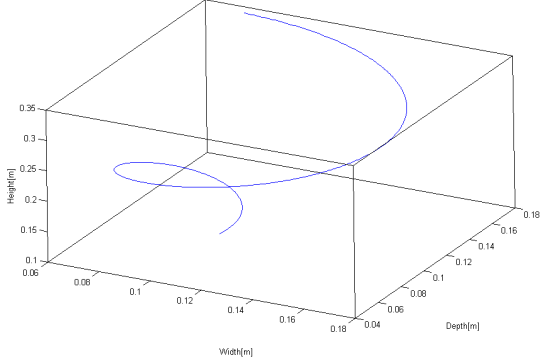


Fig. 5. Test Trajectory

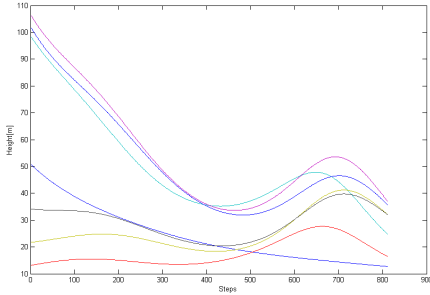


Fig. 6. Minimum Forces

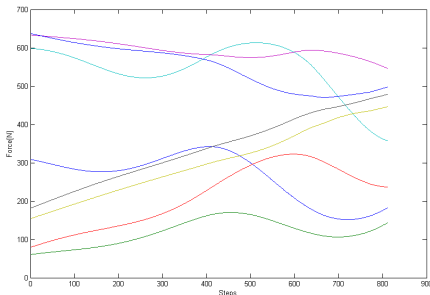


Fig. 7. Safe Forces

In fig. 6 one can see the result obtained from a standard optimizer using a NAG routine [11]. As expected the optimizer delivers tendon force distributions close to the lower

force limit. It should be noted that this implies the danger of loose tendons due to numerical effects. The tendon force distribution obtained from the proposed algorithm is shown in fig. 7. It can be seen that the tendon forces do not get close to the force limits and thus they are leading to a more tensed system. Furthermore, the calculated tendon forces are continuous and thus feasible for control.

IV. CONTINUITY OF SOLUTION

In this section the continuity of the developed algorithm in the p-norm $\|\cdot\|_p$ ($p \neq 1, \infty$), i.e. the function $\Gamma : \mathbb{R}^{m \cdot n} \mapsto \mathbb{R}^n$, which maps a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ (considered as a vector in $\mathbb{R}^{m \cdot n}$) onto the center of gravity as described before, is continuous on the set of points of the controllable workspace.

Proof: First Γ is splitted into $\text{Ker} : \mathbb{R}^{m \cdot n} \mapsto \mathbb{R}^{n \cdot r}$, which maps a matrix \mathbf{A} on its kernel \mathbf{H} represented as a vector in $\mathbb{R}^{n \cdot r}$, although it is still denoted with \mathbf{H} in calculations for simplicity, and $\text{Grav}^C : \mathbb{R}^{n \cdot r} \mapsto \mathbb{R}^n$, which maps a vector from $\mathbb{R}^{n \cdot r}$ on the center of gravity of F . Two claims have to be proven:

- 1) Ker is continuous
- 2) Grav^C is continuous

Because of $\Gamma = \text{Grav}^C \circ \text{Ker}$ the two claims finish the proof. The proof of the second claim is shown first. $\Lambda \neq \emptyset$ is assumed (i.e. the intersection of hypercube C and subspace S is non-empty and thus also the CoG exists), since continuity inside of C is to be proven. The CoG λ_s considered

$$\lambda_s^i = \frac{\int \lambda^i d\Lambda}{V(\Lambda)} \quad i = 1 \dots r \quad (21)$$

Let $\tilde{\lambda}_s$ be the CoG of $\tilde{\Lambda}$, where $\tilde{\Lambda}$ is the preimage of \tilde{F} , which is obtained from $\tilde{\mathbf{H}} = \mathbf{H} + \mathbf{E}$. The matrices \mathbf{H} and \mathbf{E} are considered (having $n \times r$ matrix entries) as vectors in

$$\mathbb{R}^{n \cdot r}, \text{ so the p-norm of } \mathbf{H} \text{ is } \left\| \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n \cdot r} \end{bmatrix} \right\|_p.$$

$$\begin{aligned} & \lim_{\|\mathbf{E}\|_p \rightarrow 0} \left| \tilde{\lambda}_s^i - \lambda_s^i \right| \\ &= \lim_{\|\mathbf{E}\|_p \rightarrow 0} \left| \frac{\int \lambda^i d\tilde{\Lambda}}{V(\tilde{\Lambda})} - \frac{\int \lambda^i d\Lambda}{V(\Lambda)} \right| \\ &= \lim_{\|\mathbf{E}\|_p \rightarrow 0} \left| \frac{V(\Lambda) \int_{\tilde{\Lambda} \setminus \Lambda} \lambda^i d\Lambda + V(\tilde{\Lambda}) \int_{\Lambda \setminus \tilde{\Lambda}} \lambda^i d\Lambda}{V(\tilde{\Lambda})V(\Lambda)} \right| \quad i = 1 \dots r \end{aligned}$$

Since the vertices of the polyhedron $\tilde{\Lambda}$ are obtained from the inequality

$$f_{\min} [1 \dots 1]^T \leq \tilde{\mathbf{H}}\boldsymbol{\lambda} \leq f_{\max} [1 \dots 1]^T \quad (22)$$

$$\Leftrightarrow f_{\min} [1 \dots 1]^T \leq \mathbf{H}\boldsymbol{\lambda} + \mathbf{E}\boldsymbol{\lambda} \leq f_{\max} [1 \dots 1]^T$$

and the vertices of the polyhedron Λ are obtained from (4), it is obvious that

$$\begin{aligned}\lim_{\|E\|_p \rightarrow 0} V(\tilde{\Lambda} \setminus \Lambda) &= 0 \\ \lim_{\|E\|_p \rightarrow 0} V(\Lambda \setminus \tilde{\Lambda}) &= 0\end{aligned}$$

and therefore

$$\lim_{\|E\|_p \rightarrow 0} \left| \tilde{\lambda}_s^i - \lambda_s^i \right| = 0 \quad i = 1 \dots r \quad (23)$$

because $\tilde{\Lambda}$ and Λ are bounded. This yields together with (7)

$$\begin{aligned}& \lim_{\|E\|_p \rightarrow 0} \left| \mathbf{x}_s^i - \tilde{\mathbf{x}}_s^i \right| \\&= \lim_{\|E\|_p \rightarrow 0} \left| \mathbf{H}_{i,r} \lambda_s - \tilde{\mathbf{H}}_{i,r} \tilde{\lambda}_s \right| \\&= \lim_{\|E\|_p \rightarrow 0} \left| \mathbf{H}_{i,r} \lambda_s - (\mathbf{H} + \mathbf{E})_{i,r} \tilde{\lambda}_s \right| \\&= \lim_{\|E\|_p \rightarrow 0} \left| \mathbf{H}_{i,r} (\lambda_s - \tilde{\lambda}_s) - \mathbf{E}_{i,r} \tilde{\lambda}_s \right| = 0 \quad i = 1 \dots r,\end{aligned}$$

This implies the second claim.

The first claim follows from the fact that the solution of a full ranked linear system of equations depends continuously on the coefficient matrix.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper an algorithm for the tendon force calculation of tendon based parallel manipulators is presented which does not contain iterative steps. Its continuity along a trajectory in the workspace is proven. Therefore it is usable within a realtime controller. Since the obtained tendon force distribution lies in the barycenter of the manifold F of the valid tendon force distributions, the solution is robust against modeling and rounding errors. As a drawback, this algorithm is computationally intensive for high redundancies, since the space wherein the triangulation is performed, has dimension of the redundancy. However, on a standard windows PC the calculations of section III-D run nearly twice as fast as a standard least squares optimizer approach using a routine from the NAG library.

B. Future Works

Due to limited time the algorithm has not been tested on the testbed yet. In the near future a force sensor equipment will be installed. Also the proposed algorithm will be implemented on the realtime target hardware. Finally, appropriate methods have to be examined to merge position and force control.

ACKNOWLEDGEMENTS

This work is supported by the German Research Council (Deutsche Forschungsgemeinschaft) under HI370/24-1 and SCHR1176/1-2.

REFERENCES

- [1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, *et al.*, "LAPACK Users' Guide, Release 2.0," *SIAM, Philadelphia*, vol. 324, 1995.
- [2] P. Bosscher and I. Ebert-Uphoff, "Wrench-based analysis of cable-driven robots," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pp. 4950–4955, 2004.
- [3] T. Bruckmann, A. Pott, and M. Hiller, "Calculating force distributions for redundantly actuated tendon-based Stewart platforms," in *Advances in Robot Kinematics - Mechanisms and Motion*, J. Lenarcic and B. Roth, Eds., Advances in Robotics and Kinematics 2006. Ljubljana, Slovenia: Springer Verlag, Dordrecht, The Netherlands, 2006, pp. 403–413.
- [4] P. Cignoni, C. Montani, and R. Scopigno, "Dewall: A fast divide and conquer delaunay triangulation algorithm in ed," *Computer-Aided Design*, vol. 30, no. 5, pp. 333–341, 1998.
- [5] S. Fang, "Design, modeling and motion control of tendon-based parallel manipulators," Ph.D. Dissertation, Gerhard-Mercator-University, Duisburg, Germany, 2005, fortschritt-Berichte VDI, Reihe 8, Nr. 1076, Düsseldorf.
- [6] P. C. Hammer, O. P. Marlowe, and A. H. Stroud, "Numerical integration over simplexes and cones," *Math. Tables Aids Comp.*, vol. 10, no. 55, pp. 130–137, July 1956.
- [7] M. Hiller, S. Fang, C. Hass, and T. Bruckmann, "Analysis, realization and application of the tendon-based parallel robot segesta," in *Robotic Systems for Handling and Assembly*, ser. International Colloquium of the Collaborative Research Center SFB 562, P. Last, C. Budde, and F. Wahl, Eds., vol. 2, Aachen. Braunschweig, Germany: Shaker Verlag, May 2005, pp. 185–202.
- [8] M. Hiller, S. Fang, S. Mielczarek, R. Verhoeven, and D. Frantiza, "Design, analysis and realization of tendon-based parallel manipulators," *Mechanism and Machine Theory*, vol. 40, 2005.
- [9] M. Husty, S. Mielczarek, and M. Hiller, "A redundant spatial Stewart-Gough platform with a maximal forward kinematics solution set," *Proceedings of the ARK '02, 8th. International Symposium on Advances in Robot Kinematics*, 2002.
- [10] A. Ming and T. Higuchi, "Study on multiple degree of freedom positioning mechanisms using wires, part 1 - concept, design and control," *International Journal of the Japan Society for Precision Engineering*, vol. 28, pp. 131–138, 1994.
- [11] NAG Ltd., "The NAG C Library Manual - Mark 8," 2007.
- [12] M. Nahon and J. Angeles, "Real-time force optimization in parallel kinematics chains under inequality constraints," in *IEEE International Conference on Robotics and Automation*, Sacramento, 11.-14. April 1991, pp. 2198–2203.
- [13] S. R. Oh and S. K. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive tensions," in *IEEE Transactions on Robotics*, 2005.
- [14] Y. Ou and L. Tsai, "Kinematic synthesis of tendon-driven manipulators with isotropic transmission characteristic," *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF MECHANICAL DESIGN*, vol. 115, pp. 884–884, 1993.
- [15] R. Verhoeven, "Analysis of the workspace of tendon-based stewart platforms," Ph.D. dissertation, University of Duisburg-Essen, 2004.
- [16] P. Voglewede and I. Ebert-Uphoff, "On the connections between cable-driven robots, parallel manipulators and grasping," in *IEEE International Conference on Robotics and Automation*, vol. 5. New Orleans: IEEE, April/May 2004, pp. 4521–4526.