A new force calculation algorithm for tendon-based parallel manipulators

Tobias Bruckmann, Lars Mikelsons, Manfred Hiller, Dieter Schramm

A New Force Calculation Algorithm for Tendon-Based Parallel Manipulators

Tobias Bruckmann*   Lars Mikelsons*   Manfred Hiller*   Dieter Schramm*

*Chair for Mechatronics, University Duisburg-Essen, Duisburg, Germany

Fig. 1.   SEGESTA Testbed

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

Abstract— **Completely and redundantly restraint tendon-based Stewart platforms demand for an appropriate distribution of tendon forces to control the platform on a given trajectory. Thus, position control has to be extended by a tendon force controller which generates continuous and feasible force values. The computation of such force distributions can be formulated as a constrained optimization problem. Solving the problem is numerically expensive and requires an algorithm which is capable to be integrated into a realtime environment. In this paper, a new algorithm for tendon force distribution calculations capable for usage on a realtime system is proposed.**

Index Terms— **Force Calculation, Tendon-based Manipulator**

I. INTRODUCTION

At the Chair of Mechatronics, a testbed for tendon-based Steward-platforms (SEGESTA <u>Se</u>ilgetriebene <u>Ste</u>wart-Platt-formen in <u>T</u>heorie und <u>A</u>nwendung) 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]. In a future modified version of SEGESTA it is planned to add an eighth tendon.
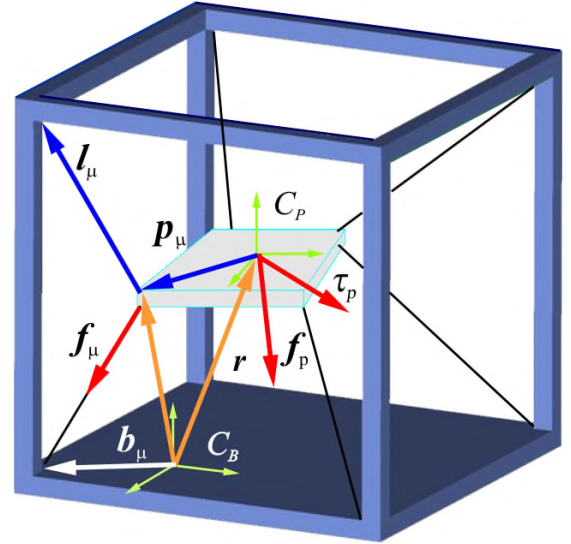
The platform can be basically guided using position control in the domain of tendon lengths. Following a trajectory, intermediate poses for every time step are calculated. For these points, the inverse kinematics delivers the corresponding tendon lengths. Since the actual tendon lengths are available from sensors, feedback control is used to guide the platform. This basic control concept provides satisfying results at low velocities. For higher accelerations and velocities it was observed that the platform begins to "wobble" due to slack tendons. To prevent slackness and also to limit forces, tension has to be controlled within lower and upper bounds. The calculation of a force distribution is theoretically straight forward in the case of a manipulator with $m = n+1$ tendons. In the case of $m > n+1$, normally optimization is used, which is always expensive in terms of computational time. For 6 d.o.f. systems, one cannot precalculate force distributes for all poses. Therefore, it is required to determine force distributions online. The chosen algorithm needs to be suitable with respect to calculation time and possibly fulfill deterministic realtime requirements.

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. I). 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].

Platform poses are calculated along a trajectory and for every step, computation of the tendon lengths (inverse kinematics) is trivial compared to the generally complicated forward kinematics. SEGESTA can be described using the following vectors and coordinate frames, with $\mu = 1, \ldots, m$ [5]:

- The coordinate frame C_B is the base frame, while C_P is connected to the platform (fig. 2).
- The vectors $\boldsymbol{b}_\mu$ denote the positions of the winch points, represented by the points where tendons are led through small ceramic eyes which are fixed.
- $\boldsymbol{p}_\mu$ are the platform-fixed vectors to the connecting points
- $\boldsymbol{l}_\mu$ denote the tendon vectors from the platform to the winches.
- The forces in the tendons are described by f_μ, where $\boldsymbol{f}_P$ and $\boldsymbol{\tau}_P$ denote all other applied forces and torques acting on the platform.

Since tendons can only transmit pulling forces, tensions must always be greater than zero which leads to the requirement of at least $m = n + 1$ tendons if no external load is available to tighten the tendons. The force equilibrium for the platform can be easily expressed as ([9],[13])

$$
\begin{bmatrix}
\boldsymbol{\nu}_1 & \cdots & \boldsymbol{\nu}_m \\
\boldsymbol{p}_1 \times \boldsymbol{\nu}_1 & \cdots & \boldsymbol{p}_m \times \boldsymbol{\nu}_m
\end{bmatrix}
\begin{bmatrix}
f_1 \\
\vdots \\
f_m
\end{bmatrix}
+
\begin{bmatrix}
\boldsymbol{f}_p \\
\boldsymbol{\tau}_p
\end{bmatrix}
= 0 \quad (1)
$$

with $\boldsymbol{\nu} = \dfrac{\boldsymbol{l}_\mu}{|\boldsymbol{l}_\mu|}$ and $\boldsymbol{f} > 0$ or in a more compact form as

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

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 we have force redundancy in the examined systems with $m \geq n + 1$ and thus at least an one-dimensional solution set for the force distributions belonging to a specific position, methods were proposed which ensure the continuity of the force distribution along a trajectory using a cost function and linear constraints for the force limits. The resulting formulation of the problem has been solved using optimization methods ([13],[11], [3], [14], [2]). Standard optimizer implementations ([1], [10]) require iterative computations which may not be used within a realtime controller system due to their normally non-predictable worst-case runtime. [12] transforms the problem to a linear programming problem which allows a direct calculation the minimum force distribution (which does not guarantee continuity) or uses iterative quadratic programming to get continuous solutions.

In this paper, we propose a non-iterative algorithm which provides continuous force distributions furthermost from the force limits. The algorithm provides a force distribution which

leads to a fairly tensed system. Due to these properties, we call the obtained solution 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 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 [12].

A. Introducing Example

As an introducing simple example, a planar manipulator having a point-shaped platform with 1 translational d.o.f. ($n = 1$) and 3 tendons ($m = 3$) is considered. It has a structure matrix with a 2-dimensional kernel. The resulting system with redundancy $r = m - n = 2$ might be limited for practical use, but it illustrates the problem very well since it allows to visualize the algorithm in three dimensions as follows:

The limits for minimum tendon force $f_{\min}$ and maximum tendon force $f_{\max}$ form an (open) cube where only forces inside the cube are acceptable, i.e. if $\boldsymbol{f}$ is a solution, it holds $\boldsymbol{f}_{\min} < \boldsymbol{f} < \boldsymbol{f}_{\max}$. Assume the force limits $f_{\min} = 1, f_{\max} = 10$. We first consider the homogenious case $w = 0$ which will afterwards be extended to the inhomogenious case which results in the simplified equations

$$
\boldsymbol{A}^\mathrm{T} \boldsymbol{f} = \boldsymbol{0} \quad (3)
$$

and

$$
\boldsymbol{f} > 0. \quad (4)
$$

The solutions of eqn. 3 are the base vectors of the kernel $\boldsymbol{H} = (\boldsymbol{H_1} \ \boldsymbol{H_2}) \in \mathbb{R}^{3 \times 2}$ of the structure matrix $\boldsymbol{A}^\mathrm{T}$. Thus, all solutions can be represented by a plane in $\mathbb{R}^3$ which is spanned by the column vectors in $\boldsymbol{H}$.

$$
\boldsymbol{f} = \boldsymbol{H} \boldsymbol{\lambda}, \quad \boldsymbol{\lambda} \in \mathbb{R}^2 \quad (5)
$$

If solutions of eqn. 3 and eqn. 4 exist, the cube of the force limits and the plane intersect and form a 2-dimensional convex polyhedron in $\mathbb{R}^3$ which contains all acceptable solutions, i.e. solutions where no tendon exceeds its minimum or maximum force limits. This is from now on called acceptable solution set.

Assume we have a structure matrix

$$
\boldsymbol{A}^\mathrm{T} =
\begin{bmatrix}
1 \\
-1 \\
-1
\end{bmatrix}
\quad (6)
$$

which allows us to describe the kernel as e.g.

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{7}$$

Since we know the force limits for the tendons, we can set the following inequality

$$f_{\min} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \leq \boldsymbol{H}\boldsymbol{\lambda} \leq f_{\max} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{8}$$

The solution set for λ of ineqn. 8 is the convex polyhedron in $\mathbb{R}^2$ which is mapped by $\boldsymbol{H}$ onto the acceptable solution set. Now, component-wise evaluation of ineqn. 8 for both sides of the inequalities gives us 6 lines in $\mathbb{R}^2$, which bound the convex preimage of the acceptable solution set under $\boldsymbol{H}$. The line intersections give a list of points in $\mathbb{R}^2$. Only the points satisfying all inequalities are the vertices of the desired convex preimage of the acceptable solution set. Afterwards, a triangulation gives a list of n_t triangles. Triangulation in this case can be easily done for $r = 2$ by choosing one vertex and connecting all vertices to the initial one. After calculation of the center of gravity (CoG) of each triangle λ_{s_i}, $i = 1, \ldots, n_t$, the resulting CoG λ_s of the convex polyhedron can be computed via

$$\lambda_s^1 = \frac{\sum\limits_i (\lambda_{s_i}^1 \cdot A_i)}{\sum\limits_i A_i} \qquad \lambda_s^2 = \frac{\sum\limits_i (\lambda_{s_i}^2 \cdot A_i)}{\sum\limits_i A_i} \tag{9}$$

The identified CoG is finally transformed back into the $\mathbb{R}^3$ using $\boldsymbol{H}$. In our example, we start with ineqn. 8 which is

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \leq \begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}, \tag{10}$$

giving us the equations

$$\begin{aligned} 1 &= \lambda_1 + \lambda_2; & 10 &= \lambda_1 + \lambda_2 \\ 1 &= \lambda_2; & 10 &= \lambda_2 \\ 1 &= \lambda_1; & 10 &= \lambda_1 \end{aligned} \tag{11}$$

which are pairwise considered as a 2×2 systems of linear equations in $\boldsymbol{\lambda}$. The resulting points $[\lambda_1, \lambda_2]$ are $[0, 1]$, $[1, 0]$, $[-9, 10]$, $[10, -9]$, $[1, 1]$, $[9, 1]$, $[10, 1]$, $[1, 9]$, $[1, 10]$, $[0, 10]$, $[10, 0]$ and $[10, 10]$. Since only the three points $[1, 1]$, $[9, 1]$ and $[1, 9]$ satisfy all inequalities of ineqn. 10, they are the vertices of the convex preimage of the acceptable solution set. In this example, we found only one triangle forming the convex polyhedron, so that its CoG is the required CoG at $\boldsymbol{\lambda} = [\frac{11}{3}, \frac{11}{3}]^T$. The mapping onto the solution plane using the kernel $\boldsymbol{H}$ (eqn. 5) gives us the solution $\boldsymbol{x}_s = [\frac{22}{3}, \frac{11}{3}, \frac{11}{3}]^T$.

In reality, normally external loads appear; at least gravity and inertia provide a non-zero w. This case is also covered by the algorithm. In the inhomgenious case, we have to deal with eqn. 2:

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f} + \boldsymbol{w} = \boldsymbol{0}, \quad \boldsymbol{f} > 0.$$

Thus, the solution plane has the form

$$\boldsymbol{f} = \boldsymbol{p} + \boldsymbol{H}\boldsymbol{\lambda}, \quad \boldsymbol{\lambda} \in \mathbb{R}^2 \tag{12}$$

i.e. external loads shift the solution plane. $\boldsymbol{p}$ denotes a particular solution of eqn. 2. In our example, let $w = 5$, the particular solution is computed by

$$\boldsymbol{A}^T\boldsymbol{p} + \boldsymbol{w} = 0 \Leftrightarrow \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} + 5 = 0.$$

This yields

$$p_1 - p_2 - p_3 = -5.$$

Now we can freely choose a particular solution, say $p_2 = p_3 = 0$. Then one finds $p_1 = -5$ which means we found a particular solution $\boldsymbol{p} = [-5 \quad 0 \quad 0]^T$. With this knowledge, instead of moving the plane, we can move the cube of the force limits by $-\boldsymbol{p}$ which is done by substracting p on both sides of eqn. 8 before performing the algorithm. The final result for the CoG must then be transformed back by adding $\boldsymbol{p}$.

B. Safe Force Calculation

Now the simple example will be extended for the general case of an arbitrary number of tendons, i.e. for any redundancy. In the general case, we have a structure matrix $\boldsymbol{A}^T \in \mathbb{R}^{n \times m}$. This leads to a kernel $\boldsymbol{H} = (\boldsymbol{H_1} \ldots \boldsymbol{H_r}) \in \mathbb{R}^{m \times r}$. Again, we start with the homogenious case of $w = 0$ which will be extended later. In the case of m tendons, the cube describing the force limits is an m-dimensional hypercube $C \subset \mathbb{R}^m$. The plane describing all solutions becomes a r-dimensional subspace $S \subset \mathbb{R}^m$ spanned by the kernel of the structure matrix. Again, if the intersection F of the hypercube C and the subspace S is non-empty, solutions $\boldsymbol{f}$ 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 done effectively using a QR decomposition of $\boldsymbol{A}^T$. Matrix $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is orthogonal which implies that we have only to consider the trapezoidal $\boldsymbol{R} \in \mathbb{R}^{n \times m}$ to get the kernel of $\boldsymbol{A}^T$.

Again, the kernel is used as a map from the $\mathbb{R}^r$ to $S \subset \mathbb{R}^m$, i.e. for all $\boldsymbol{\lambda} \in \Lambda$, the following must hold, where Λ is the (convex) polyhedron-shaped preimage of the manifold F under the mapping $\boldsymbol{H}$:

$$f_{\min} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^{m \times 1} \leq \boldsymbol{H}\boldsymbol{\lambda} \leq f_{\max} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^{m \times 1} \tag{13}$$

In other words, since the kernel $\boldsymbol{H}$ maps the $\mathbb{R}^r$ onto the solution subspace S, it maps the polyhedron $\Lambda \subset \mathbb{R}^r$ onto the solution manifold F. In fact, we don't know Λ, but we know the limits of F in eqn. 13. Thus, we use the map to calculate the vertices of Λ again by component-wise evaluation of ineqn. 13 for both sides of the inequalities which gives us $2m$ equalities which are considered in all possible combinations as $r \times r$ systems of linear equations in $\boldsymbol{\lambda}$. Every solution is examined with respect to its compliance with all inequalities of eqn. 13. 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. Remembering the introducing example having $r = 2$, we used triangulation to divide the polyhedron Λ into triangles. Triangles in $\mathbb{R}^2$ are 2-simplexes, which is now extended to the use of r-simplexes. Since we have a polyhedron, triangulation now requires advanced techniques as it is presented in [4]. Please note that we still use the term triangulation though we handle r-simplexes. Say we have a list of n_s simplexes P^k with each having $r + 1$ vertices v_{k_j} with $k = 1 \ldots n_s$ and $j = 1 \ldots r+1$. After triangulation, we determine the volumes V^k of the simplexes by integration [6] and also their CoG $\boldsymbol{\lambda}_{s_k}$.

$$\lambda^i_{s_k} = \frac{\sum\limits_{\nu=1}^{r+1} v^i_{k_\nu}}{r+1} \quad i = 1 \ldots r \quad k = 1 \ldots n_s \quad (14)$$

which can be used to calculate the CoG $\boldsymbol{\lambda}_s$ of the polyhedron via

$$\lambda^i_s = \frac{\sum\limits_{\mu=1}^{n_s} (\lambda^i_{s_\mu} \cdot V^\mu)}{\sum\limits_{\mu=1}^{n_s} V^\mu}. \quad (15)$$

Finally, the solution is transformed back using the kernel $\boldsymbol{H}$ as a map

$$\boldsymbol{x}_s = \boldsymbol{H}\boldsymbol{\lambda}_s \quad (16)$$

where $\boldsymbol{x}_s$ is the center of the manifold F.

In the inhomogenious case, we have to consider again a non-zero load w.

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f} + \boldsymbol{w} = \boldsymbol{0}, \quad \boldsymbol{f} > 0.$$

Thus, the solution subspace S has the form

$$\boldsymbol{f} = \boldsymbol{p} + \boldsymbol{H}\boldsymbol{\lambda}, \quad \boldsymbol{\lambda} \in \mathbb{R}^r \quad (17)$$

As before, external loads shift the solution plane and $\boldsymbol{p}$ denotes a particular solution of eqn. 17. The particular solution is computed by

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{p} + \boldsymbol{w} = \boldsymbol{0} \Leftrightarrow \boldsymbol{Q}\boldsymbol{R}\boldsymbol{p} + \boldsymbol{w} = \boldsymbol{0} \quad (18)$$

First, we compute a intermediate solution $\boldsymbol{y}$ by

$$\boldsymbol{Q}\boldsymbol{y} = -\boldsymbol{w} \Leftrightarrow \boldsymbol{y} = -\boldsymbol{Q}^T\boldsymbol{w} \quad (19)$$

Finally, we get $\boldsymbol{p}$ by concerning the undetermined system

$$\boldsymbol{R}\boldsymbol{p} = \boldsymbol{y} \quad (20)$$

and freely choosing r parameters.

Instead of moving the plane, we move the cube of the force limits by $-\boldsymbol{p}$ which is done by substracting p on both sides of eqn. 13 before performing the algorithm. The final result for the CoG must be transformed back by adding $\boldsymbol{p}$.

C. Proof-of-Concept

In this section we prove that the CoG of the manifold F can be computed by calculating the CoG of the convex polyhedron.

First, the CoG of a general body can be computed componentwise as

$$x^i_s = \frac{\int\limits_F x^i \, dF}{V(F)}. \quad (21)$$

Now, the theorem for integration on manifolds states

$$x^i_s = \frac{\int\limits_\Lambda x^i \circ \boldsymbol{H}^\star \sqrt{det((\boldsymbol{DH})^{\star T}(\boldsymbol{DH})^\star)} \quad d\boldsymbol{\lambda}}{\int\limits_\Lambda 1 \circ \boldsymbol{H}^\star \sqrt{det((\boldsymbol{DH})^{\star T}(\boldsymbol{DH})^\star)} \quad d\boldsymbol{\lambda}} \quad (22)$$

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

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

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

$$x^i_s = \frac{\sum\limits_{\nu=1}^{n_s} \int\limits_{P^\nu} x^i \circ \boldsymbol{H}^\star \quad d\boldsymbol{\lambda}}{\sum\limits_{\nu=1}^{n_s} \int\limits_{P^\nu} 1 \quad d\boldsymbol{\lambda}} \quad (23)$$

Now we can insert the expression $x^i \circ H^\star = \sum\limits_{\mu=1}^{r} \boldsymbol{H}_{\mu,i}\lambda^\mu$ which results in

$$x^i_s = \frac{\sum\limits_{\nu=1}^{n_s} \int\limits_{P^\nu} \sum\limits_{\mu=1}^{r} \boldsymbol{H}_{\mu,i}\lambda^\mu \quad d\boldsymbol{\lambda}}{\sum\limits_{\nu=1}^{n_s} V^\nu} \quad (24)$$

Since $\boldsymbol{H}$ is independent from $\boldsymbol{\lambda}$, it can be moved out of the integral. In vector form we get

$$x^i_s = \frac{\left(\begin{array}{ccc} H_{1,i} & \cdots & H_{r,i} \end{array} \right)}{\sum\limits_{\nu=1}^{n_s} V^\nu} \cdot \left(\begin{array}{c} \sum\limits_{\nu=1}^{n_s} \int\limits_{P^\nu} \lambda^1 \quad d\boldsymbol{\lambda} \\ \vdots \\ \sum\limits_{\nu=1}^{n_s} \int\limits_{P^\nu} \lambda^r \quad d\boldsymbol{\lambda} \end{array} \right)$$

Because of eqn. 21, we rewrite this equation by:

$$x^i_s = \frac{\left(\begin{array}{ccc} H_{1,i} & \cdots & H_{r,i} \end{array} \right)}{\sum\limits_{\nu=1}^{n_s} V^\nu} \left(\begin{array}{c} \sum\limits_{\nu=1}^{n_s} \lambda^1_{s_\nu} \cdot V^\nu \\ \vdots \\ \sum\limits_{\nu=1}^{n_s} \lambda^1_{s_\nu} \cdot V^\nu \end{array} \right)$$

Using eqn. 15, we get

$$\begin{aligned} x^i_s &= \left(\begin{array}{ccc} H_{1,i} & \cdots & H_{r,i} \end{array} \right) \left(\begin{array}{c} \lambda^1_s \\ \vdots \\ \lambda^r_s \end{array} \right) \\ &= \left(\begin{array}{ccc} H_{1,i} & \cdots & H_{r,i} \end{array} \right) \boldsymbol{\lambda}_s \end{aligned}$$

Therefore $\boldsymbol{x}_s = \boldsymbol{H}\boldsymbol{\lambda}_s$ holds where $\boldsymbol{\lambda}_s$ denotes the CoG of Λ in $\mathbb{R}^r$.

D. Example

As an example, a tendon-based Stewart platform according to the SEGESTA testbed design with the frame dimensions $800 \times 2000 \times 1500$ mm^3 and eight tendons has to follow a screw line trajectory (see fig. 3).
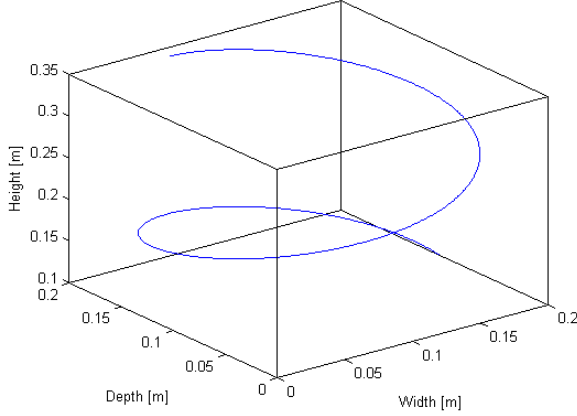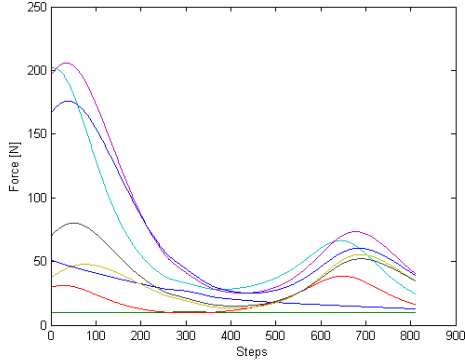
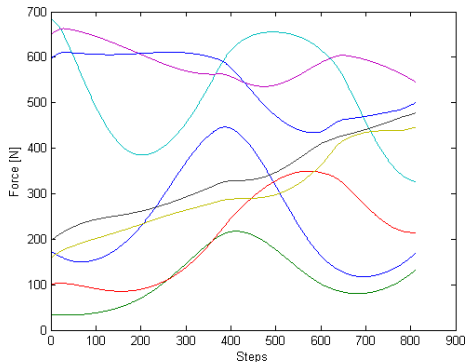

Fig. 3.   Test Trajectory

Fig. 4.   Minimum Forces

Fig. 5.   Safe Forces

The minimum force desired is $10N$, the maximum allowed force is $1000N$. The resulting minimum force distribution is shown in fig. 4, the safe counterpart is shown in fig. Obviously, the minimum force run in fig. 4 (generated by a minimizing optimizer) is close to the minimum desired force. Nevertheless, the tendon forces increase close to the workspace boundaries. Note the danger of loose tendons in case the force control doesn't work perfectly. Fig. 5 shows the force distributions generated by the save force generation. As expected, the forces remain between the given boundaries. Thus, the danger of exceeding the lower or upper force boundaries is minimized.

It is continuous and remains between the given force bounds. Thus, it is feasible for control.

E. Control Concept

The proposed algorithm was implemented for first testings on a MS Windows XP machine to calculate the results in section III-D. Compared to a standard optimizer approach using a commercial implementation (NAG), the introduced algorithm reduced the average computation time by nearly 50%. Clearly, this implementation does not allow any measurements of realtime capabilities. On the other hand, an upper limit for the required number of operations can be given since it mainly depends on the number of simplexes which is $2m - 2$ and no iterative methods are required. The implementation of the algorithm on the available realtime control system (dSPACE with Matlab/SIMULINK) is subject to current work.

Once a force distribution was calculated, the question how to apply it within a control system appears. Since both force in the tendons and the pose of the platform have to be controlled, the resulting controller has to merge both claims. The controlled devices for a tendon-based parallel manipulator are usually electric DC or EC motors, i.e. one has to set a desired motor torque via the armature voltage, so the motor torque and hence the tendon forces can be directly set. Generally, for a specific force distribution, multiple poses result in a force equilibrium. Additionally force control comes with the drawbacks of model based control, e.g. model simplifications and parameter uncertainties. Thus force control is not sufficient for a robust control. Due to this fact combinations of force control and position control are proposed. Here, mainly two concepts are presented which will be tested in the near future. Note, that the experimental evaluation is still to be done, and the presented concepts are preliminary.

1) Hybrid Force Control: The basic idea of this approach is to superpose the controller output of both the position and the force controller (fig. 6). This classical approach is well-known from force control for serial robots e.g. grinding robots. For tendon-based parallel manipulators, this approach was already introduced by [Ming1994]. For the spatial tendon-based parallel manipulator, [Fang2005] shows detailed concepts.

2) Redundant Force Control: In this paper, redundant tendon-based parallel manipulators are focused, i.e. $m \geq n+1$. Thus, the platform is fixed by $n + r$ unilateral constraints. Only n bilateral constraints are sufficient to fix the platform pose. Hence the r redundant tendons transform the n unilateral constraints into bilateral constraints, if and only if all forces in the tendon force distributions are positive.
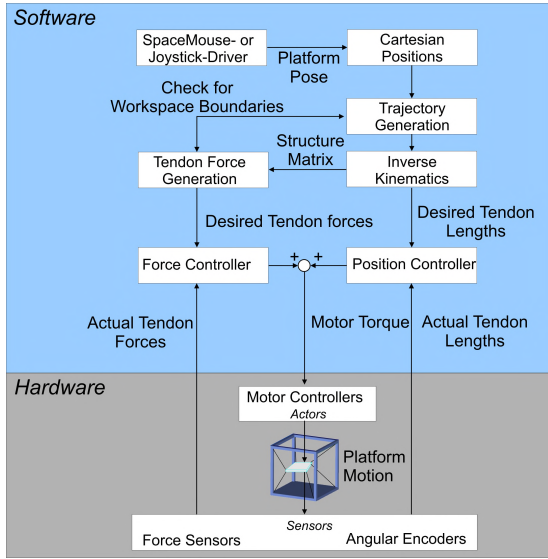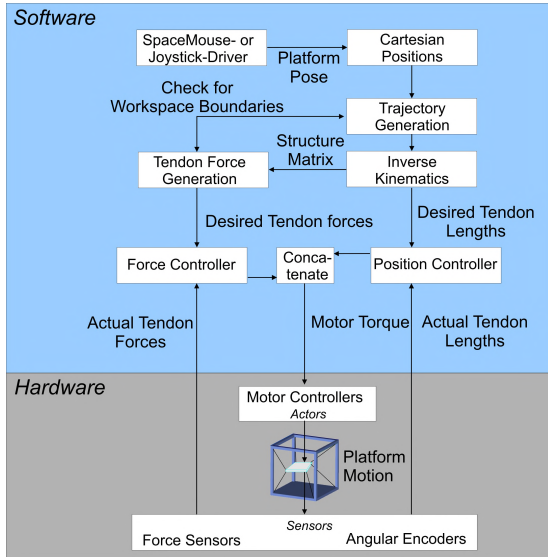
Fig. 6.   Hybrid Force Control

Fig. 7.   Redundant Force Control

The force equilibrium is represented by an underconstrained system of equations. Fixing the redundant r tendon forces results in a quadratic, uniquely solvable system of equations. Thus, the remaining n tendons react with their corresponding forces. Since the calculated force equilibrium is calculated for specific poses along a trajectory, an additional position control ensures the validity of the force calculations and precise motions of the platform (fig. 7). This approach has the additional advantage, that r tendons have to be force controlled and n tendons have to be position controlled. Thus the number of feedback sensors can be reduced to r force (or torque) sensors and n length (or angular) encoders.

IV. CONCLUSIONS

In this paper, a new algorithm for tendon force distribution calculations capable for realtime computation was shown. The algorithm delivers feasible and continuous solutions. This allows to implement the shown algorithm on realtime control systems which increases reliability, safety and performance for the industrial application of tendon-based parallel manipulators.

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." in *IEEE Int. Conf. on Robotics and Automation*, 28.-30. April, New Orleans 2004, pp. 4950–4955.

[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, Slowenien: 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, Dsseldorf.

[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. Franitza, "Design, analysis and realization of tendon-based parallel manipulators," *Mechanism and Machine Theory*, vol. 40, 2005.

[9] A. Ming and T. Higuchi, "Study on multiple degree of freedom positioning mechanisms using wires, part 1," *Int. J. Japan Soc. Prec. Eng.*, vol. 28, pp. 235–242, 1994.

[10] NAG Ltd., "NAG Fortran Library Manual, Mark 17," 1995.

[11] M. Nahon and J. Angeles, "Real-time force optimization in parallel kinematics chains under inequality constraints," in *IEEE Int. Conf. on Robotics and Automation*, Sacramento, 11.-14. April 1991, pp. 2198–2203.

[12] S. R. Oh and S. K. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive tensions," in *IEEE Transactions on Robotics*, 2005.

[13] R. Verhoeven, "Analysis of the workspace of tendon-based Stewart-platforms," Ph. D. Dissertation, Gerhard-Mercator-University, Duisburg, Germany, 2004, available under http://deposit.ddb.de/.

[14] 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.   IEEE, April/May 2004, New Orleans, pp. 4521– 4526.