IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2023, London, UK. Cite as T-RO paper.

Kinetostatic Modeling of Tendon-Driven Parallel Continuum Robots

Sven Lilge, *Student Member, IEEE,* and Jessica Burgner-Kahrs, *Senior Member, IEEE*

***Abstract*—Tendon-driven parallel continuum robots consist of multiple individual continuous kinematic chains, that are actuated in bending utilizing tendons routed along their backbones. This work derives and proposes a Cosserat rod based kinetostatic modeling framework for such parallel structures that allows for efficiently solving the forward, inverse and velocity kinetostatic problems. Using this model, the kinematic properties such as reachable workspace, singularities, manipulability and compliance of tendon-driven parallel continuum robots are studied in detail. Experiments are conducted using a real robotic prototype to validate the derived modeling approach. Overall, a median pose accuracy of 4.9 mm, corresponding to 3.4% of the continuum robots' lengths, and 6.2° is achieved. The median of the model's computation time results in 0.51 s on standard computing hardware. Fast computations of below 100 ms can be achieved, if an appropriate initial guess for solving the kinetostatic model is available, making the model suitable for a range of different applications including optimization or control.**

***Index Terms*—Continuum robots, parallel robots, flexible links, robot kinematics, robot statics.**

I. Introduction

Parallel continuum robots (PCR, example in Fig. 1), aim to overcome the drawbacks of continuum robots, such as low precision and accuracy as well as a low load capacity in comparison to conventional serial manipulators [1]. At the same time, parallel continuum robots can benefit from the inherent characteristics of continuum robots, which makes them more compliant than conventional parallel manipulators and potentially more lightweight. This enables applications in a variety of different areas, in which these characteristics are beneficial with examples ranging from minimal invasive surgery [2] to micrometer positioning [3], haptic interfaces [4] and industrial inspection and repair.

To date, different designs of PCR have been proposed and studied, proving their general feasibility. The most prominent and commonly used PCR design approach utilizes multiple continuum segments as flexible, passively bending links in a parallel setup. Rather than investigating these continuous links as actuated structures themselves, often conventional discrete joints, such as revolute or linear actuators, are used to actuate the parallel structures. The bending of the employed continuum segment then occurs passively due to external and coupling forces present in the structure. An example of this design approach is the continuum Stewart-Gough proposed in [1]. It consists of six flexible continuum links that are coupled to a common end-effector platform. The end-effector pose can be controlled by translating each of the flexible links along their principal axis, effectively changing their length. Comparable examples include planar mechanisms [5][6], mechanisms utilizing flexible continuous joints in combination with rigid links [7] and re-configurable designs [8]. Due to the simplicity of the employed continuum links, such PCR designs can achieve high position and orientation accuracies and repeatibilities [6][3], effectively inheriting the accuracy of the chosen actuators. In addition, they are easily miniaturizable making them suitable for application in highly constrained environments with limited access [2]. Several static and dynamic modeling approaches have been proposed for the resulting structures to accurately describe their deformations and motions. These include continuous formulations based on Cosserat rod theory [9][10], energy based discretized formulations [11], and pseudo-rigid body approximations [12].

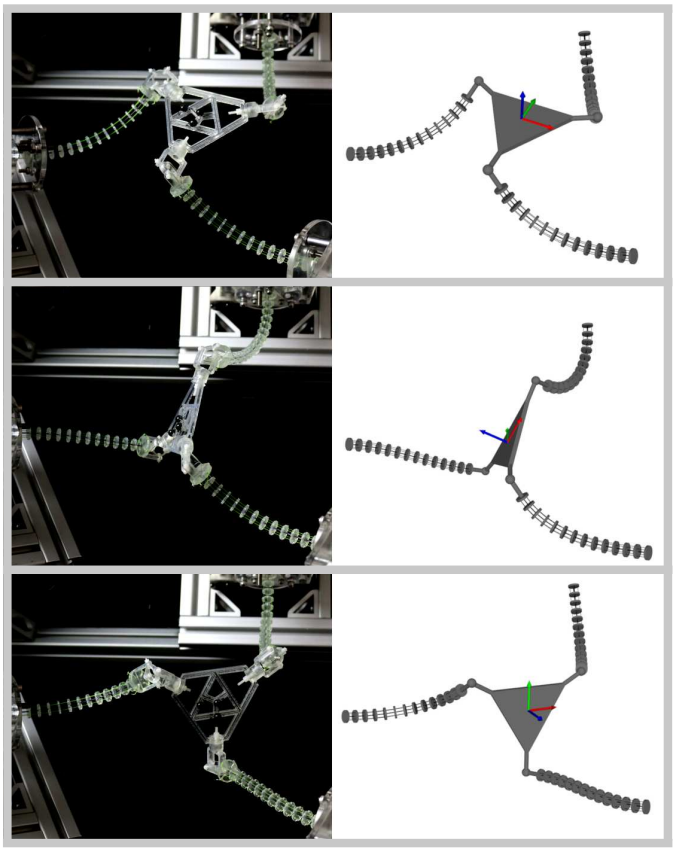

Fig. 1. Side by side view of a real and simulated tendon-driven parallel continuum robot using the proposed kinetostatic model in three different configurations.

The authors are with the Continuum Robotics Laboratory, University of Toronto, Mississauga, ON, Canada. e-mail: slilge@cs.toronto.edu.

Current research of these PCR structures mainly focuses on efficiently solving the forward and inverse static problems [13][11] - especially in the presence of multiple solutions [5] - investigating the resulting kinematic properties [9] as well as studying singularities [14] and elastic instabilities [15][16]. Example envisioned applications of such PCR include minimal invasive surgery [2][8], positioning tasks that require high accuracy [3][6], rehabilitation devices [17], and haptic interactions [4].

Contrary to the above, some PCR designs have been proposed that utilize methods to actuate and control the deformation of the flexible continuum links present in the parallel structure directly, rather than relying on external and coupling forces. This is done by employing different actuation methods commonly used in continuum robots (see [18] for an overview) to, e.g., actively bend or elongate the compliant structures. Following this approach, this paper focuses on a PCR, that makes use of tendon-driven continuum segments (see Fig. 1). In comparison to the previously discussed PCR designs, each flexible link can be considered as a fully functioning and controllable continuum robot on its own. Thus, they can potentially be studied as cooperative continuum robots that can work both independently and in a coupled parallel structure. Eventually, this might enable applications in which several continuum robots first enter a cluttered and confined environment through individual openings, before coupling into a parallel manipulator, exploiting its advantages of an increased accuracy and stiffness, e.g. as discussed in [19].

Throughout this work, we are extending our prior work on tendon-driven PCR (TDPCR) [20][21], which is limited to planar parallel robot designs. It further utilizes simplified constant curvature kinematic modeling approaches, that are solely based on geometry and are thus unable to model the statics of the manipulators. The PCR structure studied here extends our work to spatial manipulators that consist of multiple tendon-driven continuum robots (TDCR), which are coupled to a common end-effector platform, while also deriving a kinetostatic modeling approach. We designed and constructed a first prototype of such a TDPCR in [22], conducting preliminary experiments and proving the general concept of this novel type of manipulator. Related work on physically coupled, cooperative tendon-driven catheters can be found in [23], which derives a first static modeling approach. The authors extended their work to a general dynamic model for coupled TDCR in [24], evaluating it on a system consisting of two tendon-driven manipulators [25]. Both models focus on the derivation of the forward kinematic problem, i.e. mapping actuators inputs to a resulting task space shape, while the manipulators' kinematic properties such as motion capabilities, manipulability, compliance or singularities and their implications remain unexplored. Further related work focuses on the mechanical design and experimental stiffness evaluation of two cooperative TDCR, which can be controlled individual and then combine into a parallel system using an shape-memory-alloy based coupling mechanism [19].

The contributions of this paper are twofold. First, a general kinetostatic modeling framework for TDPCR is derived, extending the work in [9], which is limited to model and evaluate PCR structures with passively deforming links. The proposed framework specifically combines the modeling approach for PCR of [9] with the static modeling of TDCR presented in [26] and allows to compute both the forward and inverse kinetostatic problems of TDPCR. Additionally, the linearization of the derived boundary conditions of the robotic system are investigated, which enables efficient computation of the manipulator's Jacobian and compliance matrices as well as the identification of singularities. Second, based on the derived modeling framework, the kinematic properties of an example TDPCR design as well as their implications are studied in detail to gain general insights into the capabilities of the robot. These kinematic properties include the resulting translational workspace, the occurrences of singularities as well as the robot's manipulability and compliance in different configurations. Eventually, the derived kinetostatic model is assessed with respect to its computation time and accuracy conducting experiments on a physical robot prototype.

To the best of our knowledge, this is the first time a general kinetostatic modeling and evaluation framework is proposed for PCR featuring tendon-driven continuum links. We anticipate that it will be useful for future research on these structures on a fundamental level, laying the foundation for further investigation such as design, optimization, or control.

II. Tendon-Driven Parallel Continuum Robots

This section describes the fundamental design and structure of the proposed TDPCR, focusing on both the individual tendon-driven continuum robots and their assembly into a coupled parallel manipulator.

A. Tendon-Driven Continuum Robots

The structure of the TDCR considered in this work is shown in Fig. 2. Each robot consists of a single bending segment, keeping them relatively simple on their own. The TDCR consist of a flexible backbone of length L with a Young's Modulus E. A reference frame is attached to the TDCR base, where the z-axis is aligned with the backbone of the robot. Several spacer disks are distributed equally along the length of the robot in order to route tendons along the structure. In total up to four different tendons are considered to be routed parallel to the robot's backbone with a distance D, also called pitch radius. The tendons terminate at the last disk of the robot, where they are physically attached. Pulling these tendons allows to bend the robot into different directions. Throughout this work, we consider the tendons in the TDCR to be operated as coupled tendon pairs, meaning that while one tendon is pulled, the opposite tendon is released by the same amount. While it is generally possible to actuate each tendon individually, using only two actuators to control the tendons in pairs automatically resolves the inherent dependency between tendons in TDCR, while also requiring less motors when realizing a robotic prototype. The robot can be bent in up to two independent directions (see Fig. 2b and Fig. 2c): Controlling the first tendon pair bends the robot into the x-direction of the local base frame, while controlling the second tendon pair bends it in y-direction of the local base frame.

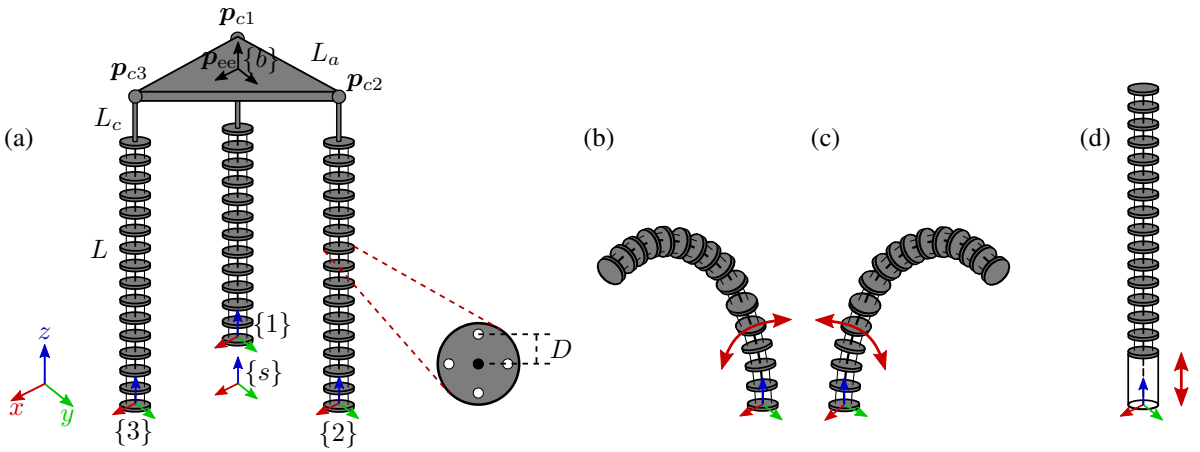

Fig. 2. (a) General structure of the proposed TDPCR consisting of three individual robots. (b-d) Different ways to actuate each of the tendon-driven continuum robots: (b) Bending in x-direction of the local base frame; (c) Bending in y-direction of the local base frame; (d) Translating the base of the robot in z-direction of the local base frame.

To further increase the workspace and dexterity of the TDCR, we introduce an additional degree of freedom in actuation. One effective way to achieve this is to consider an actuated elongation of the TDCR structure [27]. However, as this approach is challenging to implement and control, we instead are considering the translation of the base of the robot in z-direction of the local base frame as an additional actuation method (see Fig. 2d).

B. Coupled Parallel Manipulator

The arrangement of multiple TDCR into parallel structures follows some common assumptions and principles (see Fig. 2a). The designs considered in this work consist of three TDCR with identical length L, which are denoted by $i \in \{1, 2, 3\}$. Spherical joints are used to physically couple these three individual kinematic chains to a common triangular end-effector platform at positions $\boldsymbol{p}_{c1}$, $\boldsymbol{p}_{c2}$ and $\boldsymbol{p}_{c3}$. These spherical joints are located at the end of each continuum robot, considering a fixed offset of length L_c. While it is possible to couple the individual TDCR with any other joint type, we chose spherical joints to allow for less constraining motions for each chain and to minimize the internal mechanical stress present in each of them as the joints can not support coupling moments. The end-effector platform is equilateral with a side length of L_a. The world frame is denoted with $\{s\}$, while the body frame is denoted with $\{b\}$. The body frame $\{b\}$ is located in the center of the end-effector platform $\boldsymbol{p}_{\text{ee}}$. The base frames of each individual kinematic chain are denoted with $\{1\}, \{2\}$ and $\{3\}$.

As discussed in the previous subsection, each TDCR can be actuated considering different combinations of the actuation principles. To enable full pose control of the end-effector platform, six independent degrees-of-freedom (DoF) are required in actuation. For a symmetric design, each individual continuum robot is actuated with two DoF. Throughout this paper, we consider a combination of bending in one direction (e.g. along the local x-axis) as well as translation along the TDCR's principle axis as the actuation for each individual TDCR. As an additional example, redundant TDPCR designs are introduced, considering actuated bending in both the individual TDCR's local x and y direction in addition to translation. This leads to parallel manipulators with nine DoF, offering three redundant DoF in actuation.

III. Kinetostatic Modeling of TDPCR

This section focuses on deriving forward and inverse kinetostatic modeling approaches for the proposed TDPCR structures. We define the forward and inverse kinetostatic problem as the following:

$$\boldsymbol{x}_{\text{ee}} = f_{\text{FK}}(\boldsymbol{q}, \boldsymbol{w}), \tag{1}$$

$$\boldsymbol{q} = f_{\text{IK}}(\boldsymbol{x}_{\text{ee}}, \boldsymbol{w}), \tag{2}$$

where $\boldsymbol{x}_{\text{ee}} \in SE(3)$ is the robot's end-effector pose, the including position $\boldsymbol{p}_{\text{ee}} \in \mathbb{R}^3$ and orientation $\boldsymbol{R}_{\text{ee}} \in SO(3)$ of the platform's center, $\boldsymbol{w} \in \mathbb{R}^6$ is the external load, including force $\boldsymbol{f}_{\text{ext}} \in \mathbb{R}^3$ and moment $\boldsymbol{l}_{\text{ext}} \in \mathbb{R}^3$, acting on the end-effector, and $\boldsymbol{q}$ is the vector of actuation values. For actuation, we consider each tendon pair, i.e. tendons routed opposite of each other w.r.t. the backbone, to be controlled antagonistically. Specifically, we consider the tendon displacement (change in tendon length) to be actuated, where one tendon in a pair is pulled by the same amount the opposite one is released. Together with the translation, the actuation of each TDCR is expressed by two, if one tendon pair is present, or three, if two tendon pairs are present leading to a redundant design, actuation values. Thus, the actuation of the resulting parallel structure is expressed as $\boldsymbol{q} \in \mathbb{R}^6$, or $\boldsymbol{q} \in \mathbb{R}^9$ for redundant designs, including the length changes for each tendon pair as well as the translation of the TDCR bases along their local z-axes.

In order to derive the kinetostatic modeling for the parallel TDCR structures investigated in this work, we will first discuss the Cosserat rod modeling approach for each individual kinematic chain. Afterwards, boundary conditions for the resulting set of modeling equations are derived and discussed. Using the derived kinetostatic equations in combination with the boundary conditions, formulations of the forward and inverse kinematics are provided. During the derivation of our modeling equations, we mostly follow the process proposed in [9], which

is concerned with the modeling of passive link PCR. The most striking differences between our work and [9] are the inclusion of terms from [26] and [28], which allow to relate each continuum link's shape not only to external and coupling loads, but also to loads resulting from the actuation of the routed tendons.

A. Cosserat Rod Modeling of TDCR

Fig. 3 shows an overview of the geometrical and static relations of the TDPCR structure required to derive the modeling equations. Numerous approaches exist to model the individual continuum robots present in the proposed TDPCR, including both continuous and discretized approaches. We are referring to [29] for an overview and comparison of different modeling methods specifically for TDCR. Throughout this work, the individual tendon-driven continuum robots are considered and modeled as continuous Cosserat rods. These slender one-dimensional structures are mainly characterized by their state along their arc length s_i. This state includes the position $\boldsymbol{p}_i(s_i) \in \mathbb{R}^3$ and orientation $\boldsymbol{R}_i(s_i) \in SE(3)$ as well as the strain variables $\boldsymbol{v}_i(s_i) \in \mathbb{R}^3$ and $\boldsymbol{u}_i(s_i) \in \mathbb{R}^3$.

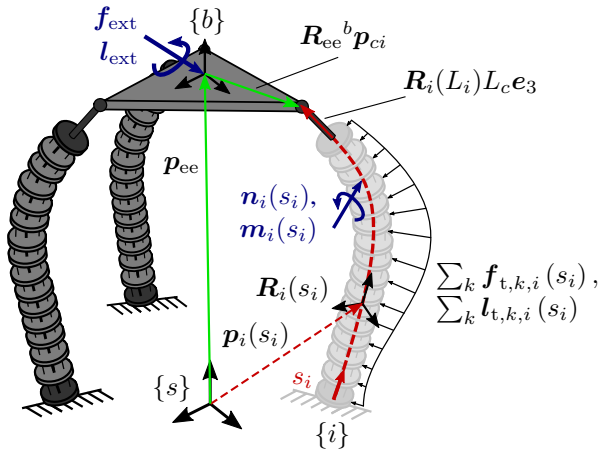

Fig. 3. Geometric and static relations for the proposed modeling approach. Each individual TDCR $i \in \{1, 2, 3\}$ is modeled as a Cosserat rod, being parameterizing its state along arc length s_i. The position of the coupling positions $\boldsymbol{p}_{ci}$ can be expressed can be described using the end-effector pose and the known geometry of the platform (green vectors) or using the resulting shape of the corresponding TDCR in combination with the fixed offset L_c (red vectors).

The strain variables describe the rate of change of $\boldsymbol{p}_i(s_i)$ and $\boldsymbol{R}_i(s_i)$, respectively, where $\boldsymbol{v}_i(s_i)$ represents the shear and elongation of the rod along its arc length s, while $\boldsymbol{u}_i(s_i)$ represents bending and twisting deformations. The position and orientation of the rod are connected to the strain variables through the rod kinematics, which are ordinary differential equations (ODE) w.r.t. arc length s_i:

$$\partial_s \boldsymbol{p}_i = \boldsymbol{R}_i \boldsymbol{v}_i, \tag{3}$$

$$\partial_s \boldsymbol{R}_i = \boldsymbol{R}_i \hat{\boldsymbol{u}}_i, \tag{4}$$

where the hat operator $\hat{\cdot}$ maps a vector from $\mathbb{R}^3$ to a skew symmetric matrix in $\mathbb{R}^{3\times3}$.

The strain variables $\boldsymbol{v}_i(s_i)$ and $\boldsymbol{u}_i(s_i)$ depend on forces and loads acting on the TDCR. Such forces and moments include distributed external loads applied to the robot's backbone, e.g. resulting from the weight of the spacer disks, as well as loads resulting from pulling and releasing the tendons routed through their structure. Each tendon $k \in \{1, ..., n_t\}$ applies forces $\boldsymbol{f}_{\text{t},k,i}(s_i)$ and moments $\boldsymbol{l}_{\text{t},k,i}(s_i)$ to the backbone along s_i, where n_t is the number of tendons per TDCR (two or four in our case, depending on the number of actuated bending directions). As outlined in [26], which describes the general static modeling approach for TDCR using Cosserat rod theory, the derivatives of the strain variables w.r.t. arc length s_i can be written as:

$$\begin{bmatrix} \partial_s \boldsymbol{v}_i \\ \partial_s \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_{\text{se},i} + \boldsymbol{A}_i & \boldsymbol{G}_i \\ \boldsymbol{B}_i & \boldsymbol{K}_{\text{bt},i} + \boldsymbol{H}_i \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{d}_i \\ \boldsymbol{c}_i \end{bmatrix}. \tag{5}$$

In this equation, $\boldsymbol{K}_{\text{se},i}$ and $\boldsymbol{K}_{\text{bt},i}$ are the robot's stiffness matrices w.r.t. shear and elongation as well as bending and torsion. Matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{G}_i$ and $\boldsymbol{H}_i$ as well as vectors $\boldsymbol{c}_i$ and $\boldsymbol{d}_i$ are simplification terms that mainly depend on the state variables $\boldsymbol{R}_i$, $\boldsymbol{p}_i$, $\boldsymbol{v}_i$, $\boldsymbol{u}_i$ and the forces and moments applied by the routed tendons. For the sake of clarity and space, we are referring to [26] for a more thorough definition and derivations of these terms. Throughout these derivations, the loads applied to the backbone resulting from tendon actuation are related to the individual tension $\tau_{k,i}$, with which a particular tendon is actuated.

In this paper, we consider tendon displacement as the more common controlled actuation in TDCR. To convert from tension to displacement and vice versa is not straightforward [28][30]. For the model derivations of this work, we are following the approach of [28] and are introducing an additional variable $\gamma_{k,i}$ that represents both tension $\tau_{k,i}$ and slack $\Theta_{k,i}$ of tendon k, as they are mutually exclusive and restricted to be positive:

$$\tau_{k,i} = \begin{cases} \gamma_{k,i}^2, & \gamma_{k,i} \geq 0 \\ 0, & \gamma_{k,i} < 0 \end{cases} \tag{6}$$

$$\Theta_{k,i} = \begin{cases} 0, & \gamma_{k,i} \geq 0 \\ \gamma_{k,i}^2, & \gamma_{k,i} < 0 \end{cases} \tag{7}$$

This variable is initially unknown and will later be solved for by considering the tendon length constraints enforced and defined by the tendon displacement actuation values in $\boldsymbol{q}$. For this, the resulting arc length $s_{k,i}$ of each tendon k along its routing path can be solved for considering the following differential equation:

$$\partial_s s_{k,i} = ||\boldsymbol{u}_i \times \boldsymbol{r}_{k,i} + \dot{\boldsymbol{r}}_{k,i} + \boldsymbol{v}_i||_2, \tag{8}$$

where $\boldsymbol{r}_{k,i}$ is a vector expressing the tendon routing location w.r.t. the TDCR's local disk frame.

Together, (3), (4), (5) and (8) represent a system of ODEs, describing the derivatives of the robot's state along arc length s_i. Solving this system of differential equations allows to obtain the state, and therefore shape, of each TDCR. The equations can be solved as an initial value problem (IVP) using numerical integration schemes. However, not all of the rod's state initial values are generally known. While the rods' initial positions $\boldsymbol{p}_i(0)$ and orientations $\boldsymbol{R}_i(0)$ are usually known from

the TDCR assembly (or depend on the actuation $\boldsymbol{q}$ in the case of a linear translation), the initial strain variables $\boldsymbol{u}_i(0)$ and $\boldsymbol{v}_i(0)$ remain unknown. Thus, they have to be solved for w.r.t. the system's boundary and constraint conditions, together with the slack/tension variable $\gamma_{k,i}$ of each tendon.

B. Constraints and Boundary Conditions of TDPCR

The TDPCR's constraints and boundary conditions used to solve for the unknown values are discussed throughout the following.

1) Static Equilibrium: First, as we are considering the TDPCR in a non-moving static configuration, we consider both a force and moment equilibrium at the center of the end-effector platform $\boldsymbol{p}_{\mathrm{ee}}$. The force equilibrium results in the following expression:

$$\left(\sum_i \left[\boldsymbol{n}_i\left(L_i\right) - \sum_k \boldsymbol{f}_{\mathrm{t},k,i}\left(L_i\right)\right]\right) - \boldsymbol{f}_{\mathrm{ext}} = \boldsymbol{0}, \tag{9}$$

where $\boldsymbol{n}_i(s_i)$ is internal force of the corresponding TDCR. It can be obtained together with the internal moment $\boldsymbol{m}_i(s_i)$ from the inner strains $\boldsymbol{v}_i(s_i)$ and $\boldsymbol{u}_i(s_i)$ using the constitutive material law:

$$\boldsymbol{n}_i = \boldsymbol{R}_i \boldsymbol{K}_{\mathrm{se},i}(\boldsymbol{v}_i - \boldsymbol{v}_i^*), \tag{10}$$

$$\boldsymbol{m}_i = \boldsymbol{R}_i \boldsymbol{K}_{\mathrm{bt},i}(\boldsymbol{u}_i - \boldsymbol{u}_i^*), \tag{11}$$

where $\boldsymbol{v}_i^*(s_i)$ and $\boldsymbol{u}_i^*(s_i)$ are the strain variables of the robot in an initially undeformed and stress free reference state. The first expression of the force equilibrium considers the internal and applied tendon forces of each individual continuum robot at their distal end $s_i = L_i$. On top of that, the external force $\boldsymbol{f}_{\mathrm{ext}}$ acting on the end-effector platform is taken into account.

Similarly, the moment equilibium can be written as:

$$\left(\sum_i \left[\boldsymbol{R}_{\mathrm{ee}}{}^b\boldsymbol{p}_{ci}\right] \times \left[\boldsymbol{n}_i\left(L_i\right) - \sum_k \boldsymbol{f}_{\mathrm{t},k,i}\left(L_i\right)\right]\right) - \boldsymbol{l}_{\mathrm{ext}} = \boldsymbol{0}. \tag{12}$$

This expression considers the moments applied to the center of the end-effector resulting from the forces, i.e. the internal force and the tendon forces, acting on the coupling position of each continuum robot. We note, that this expression does not include the moments acting on the coupling position of each TDCR as they vanish in the spherical joint and cannot be applied to the end-effector platform (see (14) for the resulting constraint equation). Both the force and moment equilibrium equations are of dimension 3, resulting in 6 boundary condition equations.

2) Spherical Joint Constraints: Next, we have to consider the constraints enforced by the spherical joints at coupling positions $\boldsymbol{p}_{c1}$, $\boldsymbol{p}_{c2}$ and $\boldsymbol{p}_{c3}$. The first constraint enforced by these joints, is that the position of the distal end of each continuum robot (considering the fixed link offset L_c) aligns with the coupling positions at the end-effector platform. For each spherical joint, this constraint can be written as:

$$\boldsymbol{p}_{\mathrm{ee}} + \boldsymbol{R}_{\mathrm{ee}}{}^b\boldsymbol{p}_{ci} - \boldsymbol{p}_i(L_i) - \boldsymbol{R}_i(L_i)L_c\boldsymbol{e}_3 = \boldsymbol{0}, \tag{13}$$

where $\boldsymbol{e}_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$. The first two parts of the equation depend on $\boldsymbol{x}_{\mathrm{ee}}$ and describe the vector to the coupling position $\boldsymbol{p}_{ci}$ using the pose of the end-effector platform as well as the known offsets ${}^b\boldsymbol{p}_{ci}$ expressed in the body frame $\{b\}$ defined by the platform geometry (green vectors in Fig. 3). The remaining parts of this equation depend on the state of the corresponding continuum robot and uses its distal position and orientation to express the coupling position $\boldsymbol{p}_{ci}$, while considering the rigid link offset L_c (red vectors in Fig. 3).

The second constraint enforced by the spherical joints, is that the moment in the coupling positions $\boldsymbol{p}_{ci}$ has to equal to zero. This can be expressed as:

$$\boldsymbol{m}_i\left(L_i\right) - \sum_k \boldsymbol{l}_{\mathrm{t},k,i}\left(L_i\right) + \left[-\boldsymbol{R}_i(L_i)L_c\boldsymbol{e}_3\right] \times \left[\boldsymbol{n}_i\left(L_i\right) - \sum_k \boldsymbol{f}_{\mathrm{t},k,i}\left(L_i\right)\right] = \boldsymbol{0}. \tag{14}$$

The first term of this expression considers the internal moment as well as the moments applied by the actuated tendons. The second term considers the moment resulting from the robot's internal forces and forces applied by the tendons in combination with the offset between the robot's distal end and the spherical joint position $\boldsymbol{p}_{ci}$.

Each of the two constraint equations discussed for the spherical joints is of dimension 3, resulting in 18 boundary condition equations, as three spherical joints are present in the proposed TDPCR design.

3) Tendon Lengths Constraints: Lastly, the length of the tendon routing paths are constrained based on the enforced tendon displacement actuation values. For each tendon k of continuum robot i, this constraint can be expressed as:

$$s_{k,i}(L_i) + \Theta_{k,i} - \left(1 + \frac{\tau_{k,i}}{(EA)_{k,i}}\right) L_i - \Delta\ell_{k,i} = 0, \tag{15}$$

where $(EA)_{k,i}$ is the tendon stiffness and $\Delta\ell_{k,i}$ is the change in tendon length defined by $\boldsymbol{q}$. This term considers both tendon slack and elongation. We are referring to [28] for a more thorough derivation of this expression. The tendon length constraint equation is of dimension 1, which results in 6 boundary condition equations, or 12, when considering redundant TDPCR designs.

Considering all of the boundary conditions and constraint equations stated in (9), (12), (13), (14) and (15), a resulting system of 30, or 36 for the redundant case, equations can be obtained:

$$\boldsymbol{b}\left(\boldsymbol{x}_{\mathrm{ee}}, \boldsymbol{q}, \boldsymbol{w}, \boldsymbol{\gamma}, \boldsymbol{u}_0, \boldsymbol{v}_0\right) = \boldsymbol{0}. \tag{16}$$

The expression depends on the end-effector pose $\boldsymbol{x}_{\mathrm{ee}}$, the actuation values $\boldsymbol{q}$, the external loads $\boldsymbol{w}$, the unknown slack/tension values for each tendon $\boldsymbol{\gamma}$ as well as the unknown initial strain values for each rod $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$.

C. Solving the Forward and Inverse Kinetostatic Problems

Based on the boundary condition equations (16), both the forward and inverse kinetostatic problems can be formulated, depending on which parameters and values are considered known and unknown. A summary of the known and unknown

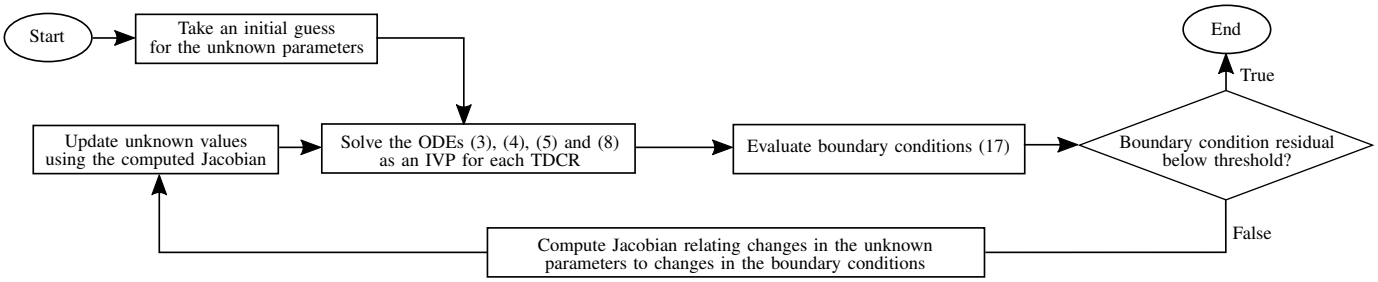

Fig. 4. Overview of the shooting method to solve the forward and inverse kinetostatic problem.

parameters for each problem, together with their dimensions as well as the number of boundary condition equations is provided in Table I. Both problem formulations result in a square system of 30 unknown parameters with 30 equations for the case of non-redundant TDPCR designs. In the redundant case, the forward kinematostatic problem has 36 unknown parameters and equations. While the number of equations remains at 36 for the inverse problem, the number of unknown parameters increases to 39. In this case, the inverse problem becomes non-unique and has infinite solutions due to the three additional degrees of freedom in actuation. This redundancy can possibly be resolved by defining additional constraint equations representing secondary tasks or by exploiting the Jacobian's null-space later on.

Throughout this work, we solve both the forward and inverse kinetostatic problems using a numerical optimization shooting method scheme. The procedure is detailed in Fig. 4. During this method, a Jacobian matrix, relating changes in the unknown parameters to changes in the boundary condition residuals, has to be computed. Resulting from the choice to formulate and solve the set of ODEs in (3), (4), (5) and (8) as a numerical integration problem, this Jacobian matrix cannot be expressed analytically. Thus, we are employing a finite differences approach to obtain a numerical approximation of this Jacobian. This approach relies on applying small perturbations to the unknown parameters and re-evaluating the boundary conditions. During this step, the system of ODEs has to be solved again. We note, that depending on which unknown parameter has been perturbed, only the ODEs for the corresponding TDCR (or even none at all) have to be recomputed. This approach has first been discussed for a continuous Stewart-Gough platform consisting of six elastic rods in [13]. We apply this method for a more efficient computation of the finite differences Jacobian of our shooting method.

We note that although the finite differences approach provides a straightforward way to approximate the Jacobian matrix of the boundary conditions, it also comes with the drawback of relatively high computation times. As an alternative, a derivation propagation approach, as presented in [31], could be implemented, which is computationally more efficient, as no re-evaluation of the boundary conditions is required.

Lastly, we would like to emphasize, that the solution of the employed shooting method heavily relies on the chosen initial guess for the unknown parameters. Generally, a default guess can be used, e.g. in which the end-effector pose lies within the origin of the world frame $\{s\}$ and each TDCR is straight, resulting in no strain at the proximal end of each TDCR and tendon tension and slack values of zero. Alternatively, a previously known TDPCR state can be used as an initial guess, e.g. when solving the model for a series of successive TDPCR configurations, which might lead to faster convergence.

IV. Linearization of the Kinetostatic Model

This section discusses the linearization of the kinetostatic model equations derived in the previous section. This linearization allows for both a straightforward derivation of Jacobian and compliance matrices [9] as well as the determination of singularities [14]. Both of these have been previously derived and studied in the context of PCR with passive flexible links, but are examined for TDPCR for the first time in this work.

Linearizing the boundary conditions (16) w.r.t. the individual variables yields the following expression:

$$\delta \boldsymbol{b} = \boldsymbol{B}_x \delta \boldsymbol{x}_{\mathrm{ee}} + \boldsymbol{B}_q \delta \boldsymbol{q} + \boldsymbol{B}_w \delta \boldsymbol{w} + \boldsymbol{B}_u \delta \boldsymbol{y}_u = \boldsymbol{0}, \qquad (17)$$

where $\boldsymbol{B}_x$, $\boldsymbol{B}_q$, $\boldsymbol{B}_w$ and $\boldsymbol{B}_u$ are partial derivatives, relating infinitesimal changes of each variable to corresponding changes in the boundary conditions. In this formulation, the variable $\boldsymbol{y}_u$ summarizes all of the uncontrolled variables $\{\boldsymbol{\gamma}, \boldsymbol{u}_0, \boldsymbol{v}_0\}$. We note, that while the infinitesimal changes for most variables in (17) can be defined in a straight-forward manner, the changes in the end-effector pose $\boldsymbol{x}_{\mathrm{ee}} \in SE(3)$ are expressed as a six-dimensional screw axis using the corresponding Lie algebra $\delta \boldsymbol{x}_{\mathrm{ee}} \in \mathfrak{se}(3)$. The partial derivatives are obtained numerically using a finite difference approach. Similarly to the numerical optimization problem discussed in the previous section, this step requires multiple re-evaluations of the boundary condition and therefore re-computations of the ODE system. However, again, depending on the perturbed variable, only a subset, or none, of these ODEs have to be re-computed.

TABLE I
Known and Unknown Parameters for the Forward and Inverse Kinetostatic Problems. Numbers in Brackets are for Redundant TDPCR Designs.

Parameter	$\boldsymbol{x}_{\mathrm{ee}}$	$\boldsymbol{q}$	$\boldsymbol{w}$	$\boldsymbol{\gamma}$	$\boldsymbol{u}_0$	$\boldsymbol{v}_0$	**Number of Unknowns**	**Number of Equations**
Dimension	6	6 (9)	6	6 (12)	9	9	–	–
Forward Kinetostatics	✗	✓	✓	✗	✗	✗	30 (36)	30 (36)
Inverse Kinetostatics	✓	✗	✓	✗	✗	✗	30 (39)	30 (36)

A. Jacobian and Compliance Matrices

In the following, the linearization of the modeling equations (17) can be used to derive both the robot's Jacobian matrices as well as its compliance matrix, an approach that has previously been demonstrated for PCR with passive flexible links [9]. For the Jacobian matrices, we assume that no change in the external loading $\boldsymbol{w}$ will occur, effectively setting $\delta\boldsymbol{w} = \boldsymbol{0}$. Using this and rearranging (17) yields:

$$\begin{bmatrix}\delta\boldsymbol{x}_{\mathrm{ee}}\\ \delta\boldsymbol{y}_u\end{bmatrix} = -\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}^{-1}\boldsymbol{B}_q\delta\boldsymbol{q}, \tag{18}$$

$$\begin{bmatrix}\delta\boldsymbol{x}_{\mathrm{ee}}\\ \delta\boldsymbol{y}_u\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\mathrm{FK}}\\ \boldsymbol{\Lambda}_{\mathrm{FK}}\end{bmatrix}\delta\boldsymbol{q}, \tag{19}$$

where $\boldsymbol{J}_{\mathrm{FK}}$ is the forward kinetostatic Jacobian matrix, relating changes in actuation $\delta\boldsymbol{q}$ to changes in the end-effector pose $\delta\boldsymbol{x}_{\mathrm{ee}}$. In addition, the term $\boldsymbol{\Lambda}_{\mathrm{FK}}$ relates changes in actuation $\delta\boldsymbol{q}$ to changes in the uncontrolled variables $\delta\boldsymbol{y}_u$. We note, that the matrix $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$ is always square with a size of 30×30 for non-redundant designs and 36×36 for redundant ones. Thus, it is always invertable as long as its rank is full.

The inverse kinetostatic Jacobian $\boldsymbol{J}_{\mathrm{IK}}$, relating changes in end-effector $\delta\boldsymbol{x}_{\mathrm{ee}}$ pose to changes in actuation $\delta\boldsymbol{q}$, can be obtained in the same way by rearranging (17):

$$\begin{bmatrix}\delta\boldsymbol{q}\\ \delta\boldsymbol{y}_u\end{bmatrix} = -\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}^{-1}\boldsymbol{B}_x\delta\boldsymbol{x}_{\mathrm{ee}}, \tag{20}$$

$$\begin{bmatrix}\delta\boldsymbol{q}\\ \delta\boldsymbol{y}_u\end{bmatrix} = \begin{bmatrix}\boldsymbol{J}_{\mathrm{IK}}\\ \boldsymbol{\Lambda}_{\mathrm{IK}}\end{bmatrix}\delta\boldsymbol{x}_{\mathrm{ee}}. \tag{21}$$

Here $\boldsymbol{\Lambda}_{\mathrm{IK}}$ relates changes in the end-effector pose $\delta\boldsymbol{x}_{\mathrm{ee}}$ to changes in the uncontrolled variables $\delta\boldsymbol{y}_u$. While the matrix $\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}$ is square for non-redundant TDPCR designs with a size of 30×30, it becomes non-square in the case of redundancy with a size of 36×39. In that case, the matrix cannot be directly inverted and a pseudo-inverse has to be considered.

Lastly, the manipulator's compliance matrix $\boldsymbol{C}$ can be obtained by assuming no change in actuation values, setting $\delta\boldsymbol{q} = \boldsymbol{0}$, and rearranging (17):

$$\begin{bmatrix}\delta\boldsymbol{x}_{\mathrm{ee}}\\ \delta\boldsymbol{y}_u\end{bmatrix} = -\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}^{-1}\boldsymbol{B}_w\delta\boldsymbol{w}, \tag{22}$$

$$\begin{bmatrix}\delta\boldsymbol{x}_{\mathrm{ee}}\\ \delta\boldsymbol{y}_u\end{bmatrix} = \begin{bmatrix}\boldsymbol{C}\\ \boldsymbol{\Lambda}_C\end{bmatrix}\delta\boldsymbol{w}. \tag{23}$$

In this equation, $\boldsymbol{C}$ relates changes in the external load $\delta\boldsymbol{w}$ to changes in the end-effector pose $\delta\boldsymbol{x}_{\mathrm{ee}}$ and the term $\boldsymbol{\Lambda}_C$ relates changes in the external load $\delta\boldsymbol{w}$ to changes in the uncontrolled variables $\delta\boldsymbol{y}_u$.

B. Singularities

Based on the derivations of the forward and inverse Jacobian matrices, two different types of singularities of the kinetostatic model can be identified, which are commonly known in the context of conventional parallel robots [32].

Observing (20), the first type of singularities (Type I) occurs, when the matrix $\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}$ becomes not invertable, i.e. loses rank. Practically, that means that there might be changes in both $\boldsymbol{q}$ and $\boldsymbol{y}_u$, i.e. moving actuators and changes in TDCR shapes, that lead to no end-effector motion. In that case, the motion of the end-effector might lose one or several DoF.

Observing (18), the second type of singularities (Type II) occurs, when the matrix $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$ becomes not invertable, i.e. loses rank. In that case, there might be changes in $\boldsymbol{x}_{\mathrm{ee}}$ and $\boldsymbol{y}_u$, resulting in motions of the end-effector or the individual TDCR shapes, even though the actuators are locked and the external load remains static. Practically, that means that parts of the robot's motion become uncontrollable and the robot becomes unstable. In (22) it can be seen, that this singularity directly affects the compliance matrix $\boldsymbol{C}$, effectively meaning that the robot might lose the ability to withstand external loads in at least one direction.

V. Kinematic Analysis

This section analyzes and discusses the kinematic properties of the proposed TDPCR structure. For that, an example TDPCR design is considered in simulation in order to study properties like its workspace, singularities, manipulability and compliance.

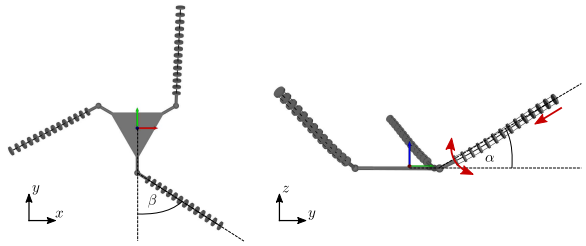

Fig. 5. Rendering of an example TDPCR design in its initial configuration $\boldsymbol{x}_{\mathrm{ee}} = \boldsymbol{I}_{4\times4}$. The three individual TDCR are tilted downwards, where α is the angle between each robot and the end-effector platform. They are further tilted in the xy-plane by an angle β w.r.t. the platform's center. Each TDCR is actuated in bending with one DoF in addition to translation along its principle axis, indicated by the red arrows.

A. Example TDPCR Design

For the example TDPCR (see Fig. 5), we are considering the design parameters stated in Table II, which corresponds to the parameters of the robotic prototype in Sec. VI. These include both geometric parameters, such as lengths and diameters, as well as physical properties, such as weights and material properties, that are required for the implementation of the derived kinetostatic modeling equations. In these equations, the backbone and disks weights are considered as distributed external loads along the robot's backbones, while the platform weight is considered as an external force $\boldsymbol{f}_{\mathrm{ext}}$ acting on the parallel robot's end-effector $\boldsymbol{x}_{\mathrm{ee}}$. For actuation of the three individual TDCR, we are considering bending with one DoF in addition to translation of each TDCR's base along its principle axis. The three individual TDCR are arranged such that they approach the end-effector platform from above with an angle of $\alpha = 30°$. In addition, they are further tilted in the xy-plane around an angle $\beta = 60°$ w.r.t. the platform's center.

For simulation, the kinetostatic model discussed in the previous sections is implemented in C++. We are solving the ODEs as an IVP using an explicit Runge-Kutta-Fehlberg (4,5) method. The numerical optimization problem to solve for the

TABLE II
PARAMETERS OF THE TDPCR DESIGNS

Segment Length L	Pitch Radius D	Number Disks	Backbone Radius r	Platform Size L_a	Coupling Offset L_c
145 mm	5 mm	13	0.5 mm	130 mm	22.5 mm
Backbone Weight m_b	**Disk Weight m_d**	**Platform Weight m_p**	**Young's Modulus E_i**	**Tendon Stiffness $(EA)_{k,i}$**	
1.2 g	0.3 g	34.4 g	54 GPa	1000 N	

system's constraints and boundary conditions is implemented using a dogleg trust region algorithm. For both of these methods we are using implementations provided by the GNU Scientific Library [33]. Our code runs on a machine with an Intel Core i7 CPU that has four 2.90GHz cores. We did not parallelize our code, which could be done by e.g. solving the IVPs of each TDCR in parallel.

B. Translational Workspace

For continuum robots, several workspace computation approaches exist, such as sampling based methods [34] or methods that compute the workspace boundaries [35], [36]. Both of these approaches have drawbacks: They either require large sample sizes leading to high computation times or do not provide additional information about singularities or manipulability. For parallel robots, one particularly studied workspace is the translational workspace, which is defined for a specific constant end-effector orientation $\boldsymbol{R}_{\text{ee}}$ and includes all reachable positions for that orientation. One common approach to obtain this workspace for parallel manipulators is the discretization method as described in [37], which we are adapting throughout this work. Such a discretization approach has been proposed for planar PCR in [38], which further makes use of an adaptive grid size.

Applying the discretization method [37] to the proposed TDPCR requires to re-compute the inverse kinematics for each grid node. That would lead to a high computational burden, assuming that a numerical optimization problem using an arbitrary initial guess has to be solved for each node. In order to overcome this, we are proposing an improved discretization approach to obtain the translational workspace. We are discretizing the three-dimensional workspace in a way that enables computation of the inverse kinematics on continuous paths. This allows us to reuse the computed values of a nearby previous inverse kinematics solution as an initial guess for the numerical optimization, leading to faster convergence. The method is illustrated in Fig. 6.

We define an initial TDPCR configuration for the workspace computation approach, in which the end-effector lies in the origin of the global frame $\{s\}$ with the orientation corresponding to the desired orientation $\boldsymbol{R}_{\text{ee}}$. We solve for this configuration using the inverse kinematics with a default initial guess, in which the individual TDCR are assumed to be straight, i.e. no initial strain and tendon slack and tension values of zero. The workspace is then discretized into n xy-planes at positions $z_1, z_2, ..., z_n$ using a step size of Δ along the z-axis. The reachable positions in each of those planes, and consequentially of the workspace, are computed as follows.

Step 1: Starting from the initial known configuration of the robot, a path is defined to approach the origin of a particular xy-plane with its z-coordinate $z = z^*$. We note, that while Fig. 6 shows xy-planes with a positive z-coordinate, z^* can be both positive or negative, allowing to define xy-planes throughout the full workspace of the TDPCR. This path is discretized into several discrete positions using the step size Δ. Using these positions and the given end-effector orientation, the inverse kinematics are then solved consecutively on every step along this path. During each step, the robot's last state is used as an initial guess. Once the desired z-coordinate has been reached, the position of the corresponding point is added to the translational workspace and the state of the robot's configuration at this point is saved. If no valid solution for the inverse kinematics can be obtained in any of the discrete steps towards $z = z^*$, the currently considered xy-plane is not part of the translational workspace. We consider the following requirements for a valid solution of the inverse kinematics: The residual of the boundary conditions has to be close to zero, i.e. smaller than some threshold ε_b, the set of obtained joint

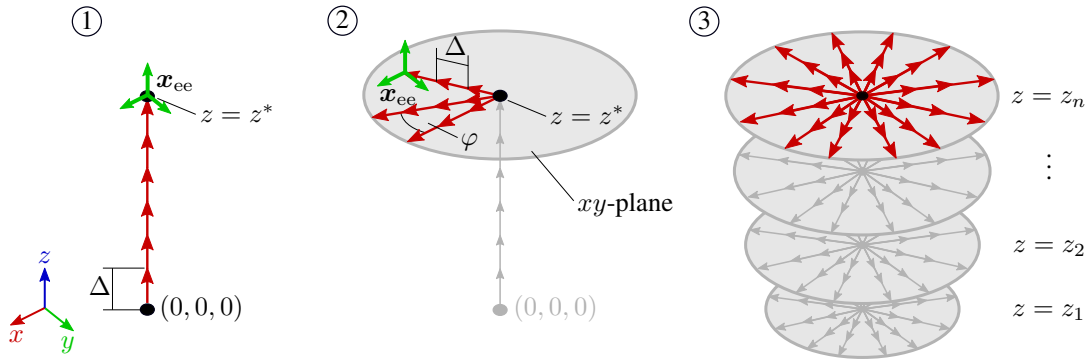

Fig. 6. Method to obtain the translational workspace of the TDPCR for a given end-effector orientation by dividing it into multiple xy-planes at positions $z_1, z_2, ..., z_n$. First, each plane is reached by continuously calculating the inverse kinematics along a discretized path from the robot's initial configuration to the origin of the plane. Second, multiple paths are defined in a circular pattern originating from the origin of each plane. These paths are again discretized and followed by continuously solving the inverse kinematics in order to obtain the boundaries of the reachable positions on that plane. Third, the reachable positions of all of the xy-planes are combined into a three-dimensional workspace.

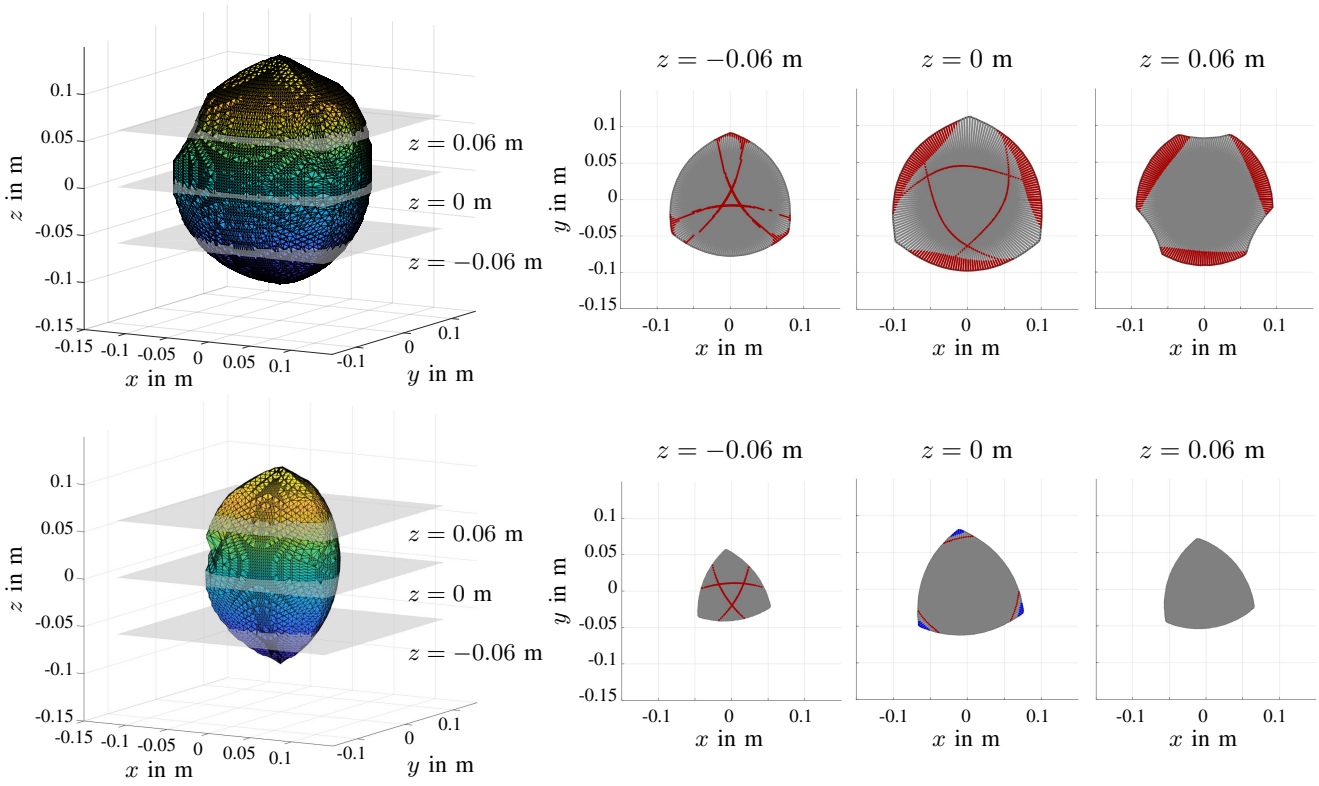

Fig. 7. Translational workspaces for end-effector orientations $\boldsymbol{R}_{\text{ee}} = \boldsymbol{I}_{3\times3}$ (top) and $\boldsymbol{R}_{\text{ee}} = \text{Rot}_z(-30°)$ (bottom) of the example TDPCR design. The overall boundaries of the three-dimensional workspaces are shown on the left, while three individual xy-planes for each workspace are shown on the right. Reachable positions are depicted in gray. Positions that lead to robot configurations that are close to Type II singularities are shown in red. Positions that are theoretically reachable, but lead to self-collisions, are shown in blue.

values has to lie within the robot's joint limits and the resulting robot configuration has to be collision-free. The process of the outlined step is visualized in ① of Fig. 6, where the green end-effector pose $\boldsymbol{x}_{\text{ee}}$ used in the inverse kinematics iterates along the discretized path in red using the step size Δ until the desired xy-plane has been reached.

Step 2: Once a certain xy-plane has been reached using the inverse kinematics, multiple paths originating from the plane's origin are considered. The paths are defined in a radial pattern using an angular offset of φ. Again, each path is discretized into multiple discrete positions using the step size Δ. Starting from the origin of the plane, the inverse kinematics are solved consecutively for each discrete position on a path considering the given end-effector orientation and using the robot's previous state as an initial guess. Every path is followed as long as valid inverse kinematics solutions can be obtained, considering the same termination criteria from above. The positions of all valid solutions are added to the translational workspace. Once an invalid solution is obtained, the robot is reset to the origin of the current xy-plane using the robot's saved state at this position. Afterwards, the procedure is repeated for the next path, which is offset by φ, until the whole xy-plane has been covered. The process of the outlined step is visualized in ② of Fig. 6. The green end-effector pose $\boldsymbol{x}_{\text{ee}}$ used in the inverse kinematics iterates along multiple discretized paths in red, which are radially arranged on the xy-plane, to identify the robot's reachable positions within that plane.

Step 3: Eventually, all reachable end-effector positions obtained from the individual xy-planes at $z_1, z_2, ..., z_n$ can be combined into a resulting three-dimensional translational workspace for the given end-effector orientation $\boldsymbol{R}_{\text{ee}}$. This step is visualized in ③ of Fig. 6.

We note, that even though multiple solutions for the same end-effector pose might exist, the proposed algorithm will only find one solution for each pose in the reachable translational workspace, depending on the initial guess used for the numerical optimization problem.

Using the outlined approach, we are calculating the translational workspace of the example TDPCR design for two different end-effector orientations, one in which the end-effector undergoes no rotation $\boldsymbol{R}_{\text{ee}} = \boldsymbol{I}_{3\times3}$ and one in which the end-effector is rotated around the z-axis $\boldsymbol{R}_{\text{ee}} = \text{Rot}_z(-30°)$. We are using a step size of $\Delta = 3$ mm and an angular offset between paths of $\varphi = 0.035$ rad. We are limiting the robot's actuation to a change of ±10 mm in tendon lengths and a translation of ±100 mm, which corresponds to approx. 70% of the TDCR length. Potential limitations of the angular range of the passive spherical joints that might result from a specific mechanical design of such a TDPCR are not considered. However, such limits could potentially reduce the workspace size and should be taken into account when computing the workspace for particular TDPCR prototypes. To obtain a collision-free workspace, we check for both collisions between the individual TDCR as well as between each TDCR and the end-effector platform. This is done by representing each disk in all of the TDCR as a sphere and calculating the

minimum distance between all spheres as well as the minimum distance between each sphere and the end-effector platform.

The obtained results are shown in Fig. 7, top for $\boldsymbol{R}_{\mathrm{ee}} = \boldsymbol{I}_{3\times3}$ and bottom for $\boldsymbol{R}_{\mathrm{ee}} = \mathrm{Rot}_z(-30^\circ)$. The computation of each workspace took approximately eight hours. On the left, the boundaries of the three-dimensional translational workspace are shown. On the right, three individual xy-planes of this workspace at $z = -0.06$ m, $z = 0$ m and $z = 0.06$ m are depicted, for which we also check the value of the reciprocal condition number for $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$, as defined in (20), to determine how close this matrix is to a singularity. We mark a particular position of the workspace as being close to a Type II singularity, if the reciprocal condition number drops below a predefined threshold $\varepsilon_s < 3.5\mathrm{e}{-9}$. When sampling these individual slices, we decreased the step size to $\Delta = 0.5$ mm to investigate the distribution of Type II singularities in greater detail. The computation of each detailed workspace slice took approximately 30 minutes. In these planes, the reachable positions together with their boundary are shown in gray. Configurations that feature a Type II singularity are shown in red. Positions that are theoretically reachable, but lead to self-collisions, are shown in blue. It can be seen that the TDPCR covers a relatively large area, allowing a wide range of motion of the end-effector in the given orientation. Nevertheless, certain areas of the workspace are subject to Type II singularities and special care might need to be taken to operate the robot close to these configurations.

C. Manipulability and Compliance

In this section, we are using the linearization of the proposed kinetostatic model to investigate the manipulability and compliance of the example TDPCR design, which provides insights into the capabilities and properties of the proposed TDPCR structures. For this, we are utilizing both the Jacobian matrix $\boldsymbol{J}_{\mathrm{FK}}$ and the compliance matrix $\boldsymbol{C}$, obtained from our kinetostatic model in (19) and (23). Throughout the following, we are considering the rotation and translation sub-blocks of the Jacobian $\boldsymbol{J}_{\mathrm{FK}}$ separately, such that we have a translational Jacobian $\boldsymbol{J}_{\mathrm{p}} \in \mathbb{R}^{3\times6}$ and a rotational Jacobian $\boldsymbol{J}_{\mathrm{o}} \in \mathbb{R}^{3\times6}$, relating changes in actuation to changes in end-effector position and orientation, respectively. Similarly, we are considering two particular sub-blocks of the compliance matrix $\boldsymbol{C}$. We define the position compliance matrix $\boldsymbol{C}_{\mathrm{p}} \in \mathbb{R}^{3\times3}$, which relates changes in the external force $\boldsymbol{f}_{\mathrm{ext}}$ to changes in the end-effector position, and $\boldsymbol{C}_{\mathrm{o}} \in \mathbb{R}^{3\times3}$, which relates changes in the external moment $\boldsymbol{l}_{\mathrm{ext}}$ to changes in the end-effector orientation.

1) Manipulability and Compliance Ellipsoids: One common approach to evaluate the manipulability of robot manipulators is to draw ellipsoids from the corresponding Jacobian matrix for specific configurations using its singular values and vectors [39]. We apply this approach to both the translational and rotational Jacobian matrices $\boldsymbol{J}_{\mathrm{p}}$ and $\boldsymbol{J}_{\mathrm{o}}$. The resulting ellipsoids indicate the robot's ability to translate along or rotate around a particular direction, where a large ellipsoid radius in a direction means that relatively small actuator displacements are required to move the robot in that direction.

The same approach can be applied to the compliance matrices $\boldsymbol{C}_{\mathrm{p}}$ and $\boldsymbol{C}_{\mathrm{o}}$ to investigate the magnitude of the robot's end-effector displacements due to external loads. Compliance ellipsoids can be defined for each of these matrices by computing the corresponding singular values and vectors. Similar to the manipulability ellipsoids, a large radius of these ellipsoids in a certain direction means that a relatively small load is required to displace the end-effector in that direction.

In the following, we are computing and visualizing all of the discussed ellipsoids for the example TDPCR design in its initial configuration $\boldsymbol{x}_{\mathrm{ee}} = \boldsymbol{I}_{4\times4}$ (see Fig 8, top row). We are further reporting on the three singular values of each matrix used to obtain the ellipsoid (in decreasing magnitude), indicating the three radii of the resulting ellipsoids in their main directions. It can be seen, that the position manipulability ellipsoid is isotropic in the x and y-directions, while it is slightly larger in the z-direction. That means, that less actuator displacements are required to move the end-effector in z than in x and y. The orientation ellipsoid on the other hand has larger radii in x and y, indicating that less actuator displacements are needed to tilt the end-effector around these axis than twisting it around z. The scale of the ellipsoids show that actuator displacements in the order of 1 mm lead to translational end-effector displacements of several millimeters as well as rotational displacements of below 1 rad. The position compliance ellipsoid shows, that smaller external forces are required to displace the end-effector in z-direction than in x and y, meaning that the robot offers a higher stiffness in the latter directions. Looking at the orientation compliance ellipsoid, smaller external moments are required to tilt the end-effector around x and y than to twist it around z, meaning that the robot can withstand higher moments in that direction and exhibits a higher rotational stiffness. The scale of the ellipsoids show that external forces in the order of 1 N lead to translational displacements of several millimeters and external moments in the order of 1 Nm lead to significant rotational displacements of several radians.

To provide a more thorough evaluation of the TDPCR's kinematic properties, Fig 8 includes manipulability and compliance ellipsoids from two additional example configurations. These include a twist around the z-axis with -45° and a rotation around the y-axis with -90°. Overall, it can be concluded, that the proposed TDPCR design is well conditioned w.r.t. the shown ellipsoids throughout different configurations. The robot's end effector can be fully manipulated and moved along the different task space directions, making it generally suitable for robotic manipulation applications.

2) Condition Number of Jacobian and Compliance: The ellipsoid-based investigations of the manipulability and compliance are useful to analyze the detailed TDPCR behavior in particular configurations. We are further analyzing the reciprocal condition numbers of both the translational and rotational Jacobian and compliance matrices to provide a more general investigation of these measures throughout the whole translational workspace for $\boldsymbol{R}_{\mathrm{ee}} = \boldsymbol{I}_{3\times3}$. The reciprocal condition numbers, i.e. the ratios of the smallest and largest singular values of the Jacobian and compliance matrices, quantify the isotropy of the corresponding ellipsoids.

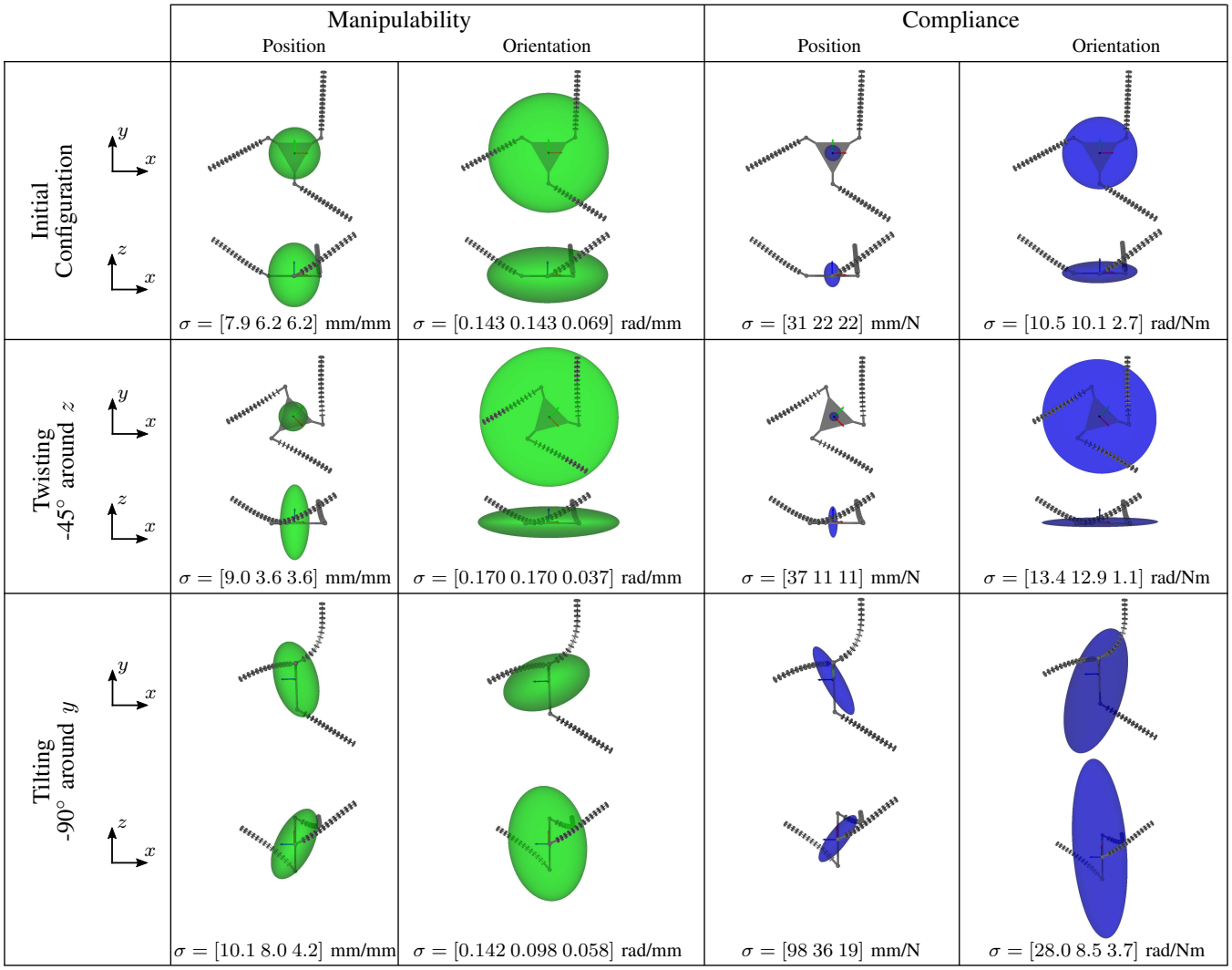

Fig. 8. Manipulability and compliance ellipsoids for three different TDPCR configurations: The robot's initial configuration $\boldsymbol{x}_{\text{ee}} = \boldsymbol{I}_{3\times3}$, twisting the end-effector around z and tilting the end-effector along y. Two views are depicted for each ellipsoids, one in the xy-plane and one in the xz-plane. In addition, the magnitudes of the three radii are stated for each ellipsoid, which correspond to the singular values of the matrix used to obtain the ellipsoid.

Tab. III reports on the mean, minimum and maximum values of these reciprocal condition numbers for the translational and rotational Jacobian and compliance matrices throughout the translational workspace with $\boldsymbol{R}_{\text{ee}} = \boldsymbol{I}_{3\times3}$. In addition, Fig. 9 (top) shows the distribution of the reciprocal condition numbers of the corresponding matrices for one workspace slice at $z = 0$m. It can be seen that each matrix is best conditioned in the center of the workspace and the reciprocal condition numbers decrease towards the edges of the workspace. Further, both the translational Jacobian and compliance matrices are generally better conditioned than the rotational ones. This can be explained by the more uniform ellispods as seen in Fig. 8. Both the rotational Jacobian and compliance matrix have overall lower reciprocal condition numbers, since less actuator displacements or external moments are required to tilt the end-effector around x and y than to twist it around z.

TABLE III
JACOBIAN AND COMPLIANCE MATRIX RECIPROCAL CONDITION NUMBERS THROUGHOUT THE TRANSLATIONAL WORKSPACE FOR $\boldsymbol{R}_{\text{ee}} = \boldsymbol{I}_{3\times3}$

	$\boldsymbol{J}_{\text{p}}$	$\boldsymbol{J}_{\text{o}}$	$\boldsymbol{C}_{\text{p}}$	$\boldsymbol{C}_{\text{o}}$
mean	0.3910	0.3563	0.2763	0.2636
max	0.9986	0.9837	0.9822	0.9793
min	3.25×10^{-4}	1.15×10^{-4}	4.59×10^{-4}	7.33×10^{-5}

D. Singularities

In the following, we are investigating the Type II singularity occurrences as seen in Fig. 7 in greater detail. For this, we are particularly focusing on the workspace slice at $z = 0$ m for an end-effector orientation of $\boldsymbol{R}_{\text{ee}} = \boldsymbol{I}_{3\times3}$. In Fig. 9 (middle) we have plotted the reciprocal condition number distributions in this workspace slice for the matrices $\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$, indicating the closeness to Type I and Type II singularities, and for matrix $\boldsymbol{B}_u$. From this, two different cases, in which Type II singularities occur, can be observed.

In the first case, only the matrix $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$ is close to loosing rank, while matrices $\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}$ and $\boldsymbol{B}_u$ exhibit no rank deficiency. This case occurs towards the edges of the workspace slice. Here, the resulting robot configuration leads to a not fully constrained end-effector platform. Due to this, the end-effector gains additional DoF, that are uncontrollable. This also means, that the end-effector loses the ability to withstand external forces or moments in certain directions. This can

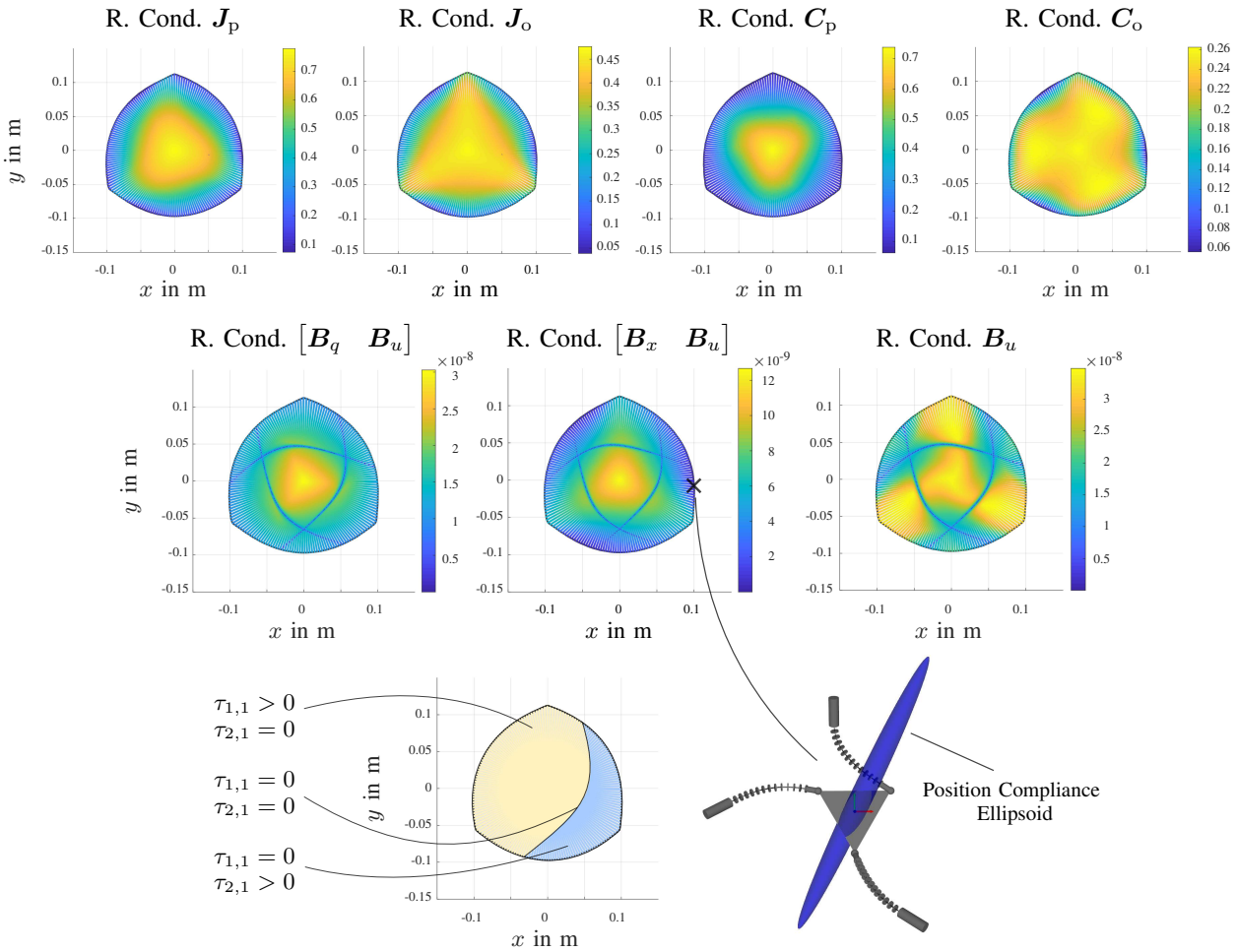

Fig. 9. Top: Reciprocal condition number distributions of the Jacobian and compliance matrices throughout the translational workspace at $z = 0$ m for $\boldsymbol{R}_{\mathrm{ee}} = \boldsymbol{I}_{3\times3}$. Middle: Reciprocal condition number distributions for $[\boldsymbol{B}_q\ \boldsymbol{B}_u]$, $[\boldsymbol{B}_x\ \boldsymbol{B}_u]$ and $\boldsymbol{B}_u$ throughout the translational workspace at $z = 0$ m for $\boldsymbol{R}_{\mathrm{ee}} = \boldsymbol{I}_{3\times3}$. Bottom left: Tendon tension distribution for one individual tendon-driven continuum link throughout the workspace slice. Bottom right: Example TDPCR configuration with the corresponding position compliance ellipsoid close to a Type II singularity.

further be seen in the plots of the reciprocal condition numbers of the compliance matrices in Fig. 9 (top), which are relatively small at the edges of the workspace. To visualize this effect, a particular TDPCR configuration close to the workspace edge together with the position compliance ellipsoid is shown in the bottom right part of Fig. 9. It can be seen, that the ellipsoid is relatively large in one direction, indicating that the end-effector is close to loosing the ability to withstand forces in this particular direction.

In the second case, the matrix $\boldsymbol{B}_u$ is close to becoming singular, which leads to the matrices $\begin{bmatrix}\boldsymbol{B}_q & \boldsymbol{B}_u\end{bmatrix}$ and $\begin{bmatrix}\boldsymbol{B}_x & \boldsymbol{B}_u\end{bmatrix}$ being close to loosing rank. These type of singularities are called leg singularities, first discussed for PCR in [14], in which there are changes in the uncontrolled variables $\boldsymbol{y}_u$ while both the actuators and end-effector exhibit no motion. Fig. 9 shows that this type of singularity occurs on three distinct lines throughout the TDPCR workspace slice, each of which can be related to one of the individual continuum links. Specifically, a leg singularity for a particular tendon-driven continuum link occurs, when both of the routed tendons are not exhibiting any tension. In the bottom left of Fig. 9, two distinct workspace slice areas can be identified, when investigating the tendon tensions for one of the three continuum links. It can be seen, that only one of the two tendons is under tension at any time (yellow area for first tendon, blue area for second tendon), while the other one exhibits slack. In the transition between these two areas, which corresponds to the leg singularity locations, both tendons exhibit no tension and matrix $\boldsymbol{B}_u$ loses rank. We note that this phenomena can potentially be overcome by applying enough pretension to both tendons, so that both of them are always under tension and slack is eliminated [28].

VI. Kinetostatic Model Assessment

In this section, the derived kinetostatic model will be assessed both with respect to its accuracy in comparison to a robot prototype as well as its computation time.

1) Robot Prototype: To evaluate the accuracy of the model, experiments with a real robotic prototype are conducted. The prototype is based on the re-configurable parallel manipulator proposed in [22]. The design parameters are updated and slightly altered to match the design investigated in the previous section, as stated in Tab. II. The three TDCR are spatially arranged according to Fig. 5 with $\alpha = 30°$ and $\beta = 60°$ to match the design investigated in simulation. To show general validity of the derived model for both non-redundant and redundant TDPCR designs, each TDCR of the prototype is actuated using two tendons pairs in addition to translating its

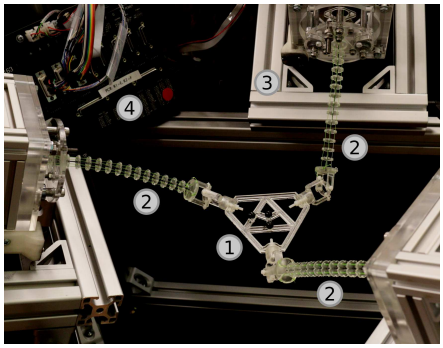

Fig. 10. TDPCR prototype consisting of a triangular end-effector platform ①, three tendon-driven continuum robots ②, controlled with actuation units ③ and motor control boards ④.

base, leading to a manipulator with nine DoF. Each actuator is realized by a motor-gearbox-combination (Part no. 595955, Maxon Motor AG, CH). Synchronous control of the motors is achieved using Galil DMC-4163 motion controllers (Galil Motion Control, USA). Fig. 10 shows a picture of the robot prototype, including its end-effector ① connected to the three TDCR ②, each with an individual adjustable actuation unit ③ and a motor controller ④.

2) Experiment Setup and Data Set: For the calibration and assessment of the kinetostatic model, a data set of thirty different robot configurations is collected, each consisting of the robot's actuation values and the resulting end-effector pose. These configurations are first identified using the simulated TDCR and are chosen so that their end-effector poses are evenly distributed, covering a large area of robot's workspace, while respecting the mechanical limits of the physical prototype. The thirty resulting end-effector poses in these configurations are visualized as frames in Fig. 11.

In order to measure the end-effector pose of the physical prototype in the given configurations, a contact-less measurement arm with a laser probe (FARO Edge with FARO Laser Line Probe HD, FARO Technologies Inc., USA) is utilized. Highly accurate and detailed point cloud readings of the robot's shape can be obtained by scanning it with the laser probe, which has a stated accuracy of $\pm 35\,\mu$m. Three spheres with diameters of 8 mm are attached to the TDPCR end-effector platform, that can be easily identified and extracted from the corresponding point cloud. Using the known positions of these sphere with respect to the end-effector platform, the frame holding the spatial position and orientation of the end-effector can be reconstructed (see Fig. 12). In addition, a world reference frame consisting of three additional spheres is defined and rigidly attached to the non-moving part of the robot prototype. Each individual TDCR base is equipped with three spheres as well, allowing to accurately identify the base frame of each. Measuring the sphere positions of both the TDCR base frames and the world frame allows for the reconstruction of their respective position and orientation with respect to the internal frame of the measurement arm system. Finally, transforming all of the measured frames into the world frame allows to align the measurement data with the kinetostatic model in simulation.

3) Model Calibration and Accuracy: For each configuration, we asses the accuracy of the kinetostatic model by

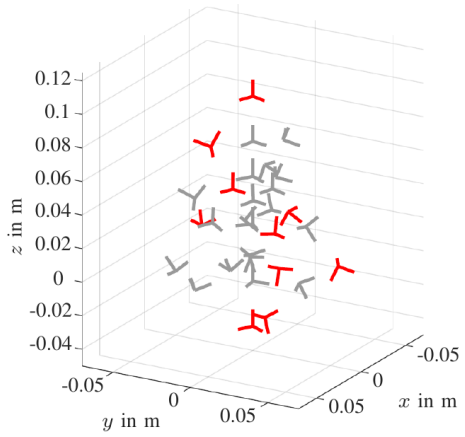

Fig. 11. End-effector poses visualized as frames for the thirty configurations in the data set. Configurations used for the model calibration process are depicted in gray and the ones used to detect and prevent overfitting in red.

comparing the computed end-effector position and orientation with the corresponding measurement. For the position, we compute the Euclidean distance between measurements and model prediction. For orientation, we are first computing the rotation between the measured and modeled frames $\Delta\boldsymbol{R} = \boldsymbol{R}_{\text{mes}}\boldsymbol{R}_{\text{model}}^{T}$. We then transform the resulting rotation matrix in an angle-axis pair, consisting of a normalized axis of rotation and the angle θ of the rotation around this axis. We are considering the absolute value of θ to evaluate the accuracy in orientation.

During the evaluation of the proposed kinetostatic model, we are employing an optimization routine, which aims to find calibrated model parameters that minimize the offset between model and measurements. The to be optimized parameter set includes the Young's modulus of the TDCR backbones E_i, the tendon stiffness $(EA)_{k,i}$ as well as small rotational offsets $\delta\alpha_i$, $\delta\beta_i$ and $\delta\gamma_i$ for each TDCR base. Each rotational offset describes a rotation around one of the elementary directions, so that the resulting rotational offset for each TDCR base is $\delta\boldsymbol{R}_i = \text{Rot}(z,\delta\alpha_i)\text{Rot}(y,\delta\beta_i)\text{Rot}(x,\delta\gamma_i)$. We further assume, that all of the TDCR backbones and tendons exhibit the same material properties. Thus the resulting parameter set consists of eleven individual values $P = \{\delta\alpha_1, \delta\beta_1, \delta\gamma_1, \ldots, \delta\alpha_3, \delta\beta_3, \delta\gamma_3, E_i, (EA)_{k,i}\}$. The to be optimized parameters are chosen due to possible uncertainties in the assumed material properties of the utilized backbones and tendons as well as possible inaccuracies during assembly of each TDCR, leading to small rotational misalignments of their bases. For optimization, we are using the Nelder-

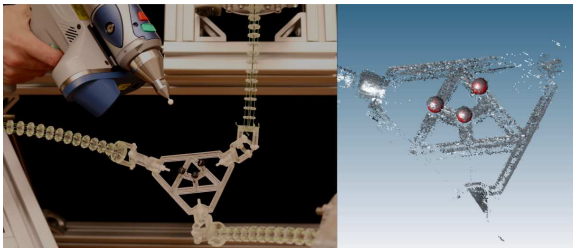

Fig. 12. Measurement of robot pose. Left: Scanning the end-effector with the a laser probe. Right: Point cloud of the scanned end-effector with extracted spheres in red.

Mead Simplex algorithm as implemented in the C++ GNU Scientific Library [33], where the cost function is the average position error between model and measurements over all of the considered configurations $1, ..., N$. The optimization problem can be written as:

$$P_{\text{calibrated}} = \arg\min_{P} \frac{1}{N} \sum_{i=1}^{N} ||\boldsymbol{p}_{\text{mes},i} - \boldsymbol{p}_{\text{model},i}||_2. \quad (24)$$

To avoid overfitting of the calibrated parameters, we are only considering a subset of twenty configurations for the optimization routine (visualized in gray in Fig. 11). During each iteration of the optimization, we are evaluating the average position error between model and measurements for the remaining ten configuration to check if the calibrated parameters lead to a generalized minimization of the position error. We are stopping the optimization process, when an overall increase in the average position error over these ten configuration can be observed, while the error of the remaining twenty configurations further decreases, indicating overfitting of the model parameters. The resulting updated model parameters obtained from the calibration process are stated in Tab. IV. Both the Young's modulus and the tendon stiffness remain mostly unchanged. The angular offsets for the TDCR bases are small and lie within ranges that can be expected from the manual manufacturing process of the manipulators. We note, that while overfitting during optimization was avoided and the calibrated model parameters show realistic values, there is generally no guarantee that they are physically correct. They may in fact compensate for other unmodeled effects or contain errors from other factors.

TABLE IV
CALIBRATED TDPCR PARAMETER VALUES

$\delta\alpha_1$	$\delta\beta_1$	$\delta\gamma_1$	$\delta\alpha_2$	$\delta\beta_2$	$\delta\gamma_2$
-4.19°	6.55°	0.28°	-8.15°	5.86°	0.94°
$\delta\alpha_3$	$\delta\beta_3$	$\delta\gamma_3$	E_i	$(EA)_{k,i}$	
-5.37°	2.96°	4.20°	54.15 GPa	983.60 N	

The resulting modeling accuracies before and after calibration can be found in Tab. V. Using the initial uncalibrated parameters of the proposed model results in a median position error of 8.5 mm, which corresponds to 5.9% of the length of each TDCR, and a median orientation error of 7.2°. After calibration, median position and orientation errors result of 4.9 mm, corresponds to 3.4% of the TDCR length, and 6.2° can be achieved. Possible reasons for the remaining errors might result from inaccuracies in the 3D printed parts of the physical prototype, an inaccurate calibration of the actuators' zero positions, which has been performed manually based on visual perception, as well as the presence of unmodeled effects such as hysteresis of the elastic materials and friction between the tendons and the TDCR spacer disks. Theoretically, more model parameters could be calibrated to compensate for some of the aforementioned effects, however, a bigger data set consisting of more individual robot configurations should be collected for that to make the optimization problem less prone to overfitting.

4) Computation Time: Lastly, we are evaluating the computation time of the model for the thirty configurations in our data set by computing the forward kinetostatic problem for each. The computation time mostly relies on the number of iterations required for the shooting method (cf. Fig. 4) that is used to solve the boundary value problem of the kinetostatic model. We note, that the number of required iterations and computation time heavily depends on the initial guess used during that shooting method. Throughout the following, we consider that the robot state is known for the first configuration of the data set, in which all of the actuator values equal to zero. We are using this robot state as the initial guess when solving the forward kinetostatic problem for all of the thirty configurations of the data set. As stated in Sec. V-A, the calculations are computed on a machine with an Intel Core i7 CPU that has four 2.90GHz cores, of which one core was utilized.

TABLE V
POSITION AND ORIENTATION MODELING ERRORS

	Before calibration		After calibration	
	pos. in mm	ori. in °	pos. in mm	ori. in °
median	8.5 (5.9%)	7.2	4.9 (3.4%)	6.2
mean	9.5 (6.5%)	8.9	6.5 (4.5%)	6.6
std	4.5 (3.1%)	5.8	4.0 (2.7%)	3.3
max	22.4 (15.4%)	30.0	15.2 (10.5%)	16.8

The resulting number of iterations and computation times are stated in Tab. VI. On average, the model took 86.4 iterations to converge, resulting in a mean computation time of 1.53 s. It can be seen, that the median value for both the number of iterations and computation time is significantly lower with 28.5 iterations and 0.51 s, respectively. We note, that the computation time of the kinetostatic model heavily relies on the chosen initial guess of the shooting method. If the resulting robot state is similar to the assumed initial guess, fast computation times can be achieved, while a significant increase can be observed for configurations far away from that guess. Generally, when controlling and operating robotic manipulators, the robot state of a previous configuration is assumed to be known as the robot is updated continuously based on its last state.

To further quantify the kinetostatic model's computation times in such scenarios, we consider some example continuous motion sequences of the proposed TDPCR structure. We investigate two distinct motion sequences, each discretized and interpolated with several discrete configurations. The resulting state of the TDPCR during these motions is calculated for each of the discrete configurations using the last known state as an initial guess. Both motion sequences are shown in the video attachment of this paper. The resulting computation times are reported in Tab. VII The first motion sequence includes combined elementary translations and rotations of the end-effector platform for the non-redundant TDPCR design from Sec. V. It consists of 1300 individual configurations and the robot state for each is computed by solving the inverse kinetostatic problem considering the desired end-

TABLE VI
NUMBER OF MODEL ITERATIONS AND COMPUTATION TIMES

Iterations				Computation Time in ms			
median	mean	std	max	median	mean	std	max
28.5	86.4	101.4	294	508	1533	1771	5030

TABLE VII
COMPUTATION TIMES OF THE MOTION SEQUENCES

Non-redundant TDPCR			
median	mean	std	max
67.6 ms	69.4 ms	17.8 ms	533.4 ms
Redundant TDPCR			
median	mean	std	max
62.5 ms	77.5 ms	56.7 ms	953.7 ms

effector pose. The average computation time results in 69.4 ms. The second motion sequence is defined in the actuator space of the redundant TDPCR prototype design from Sec. VI and interpolates between several example configurations of the data set in Fig. 11. It consist of 1000 configurations, in which the robot state is obtained by solving the forward kinetostatic problem considering the given actuator values. The average computation time results in 77.5 ms. For both motion sequences, the changes in actuation values between two subsequent configurations are below 0.1 mm for the tendon length changes and below 1 mm for the translational actuators.

It can be concluded, that on average fast computation times below 100 ms can be achieved when solving the kinetostatic model, provided that an appropriate initial guess is available, making it suitable for a range of different applications including optimization or control.

VII. CONCLUSION

In this work, a comprehensive modeling and evaluation framework for TDPCR is proposed. It allows to solve both the forward and inverse kinetostatic problems for TDPCR, while the linearization of their boundary conditions enable efficient computation of Jacobian and compliance matrices in addition to the identification of singularities. The derived kinetostatic model shows good agreement between simulations and experiments on a physical robotic prototype, resulting in median position and orientation errors of 4.9 mm and 6.2°, respectively. Fast computation times of the model, up to below 100 ms, can be achieved if adequate initial guesses of the robot state are available. We showed, that the proposed framework can be used to investigate the relevant kinematic properties of TDPCR including their translational workspaces, singularity distributions, manipulability and compliance. Utilizing an example design, it could be shown that the TDPCR can offer well conditioned manipulability and stiffness properties in addition to large translational workspaces.

We note, that the investigation of the kinematic properties of TDPCR in this work are limited to numerical values due to the nature of the employed numerical algorithms. A possible discretization of the proposed model or the usage of an alternative approach to model the kinetostatics of the continuum links, such as a discrete psuedo rigid body model, e.g. as discussed in [29], could enable analytical formulations of the linearization of the system's boundary conditions. This could lead to analytical formulations for the Jacobian, compliance matrices and the singularity conditions, allowing for more in-depth studies and investigations. In a similar vein, we note that a relation between elastic instabilities and Type II singularities has been shown for passive link PCR using a discretization approach in [14] and future work should focus on extending these investigations to TDPCR.

In addition, the current modeling framework currently only considers TDCR with a single bending segments. Future work should focus on extending the framework to include continuum robots with multiple bending segments and investigate their implications on the capabilities and properties of the resulting parallel robot. Further, the possibilities of kinematic redundancies of the TDPCR structures should be investigated in greater detail. The inclusion and exploration of such redundancies can be advantageous in specific applications to fulfill secondary tasks in addition to controlling the robot's pose, such as stiffness optimization [40], obstacle and self-collision avoidance [41][42] or singularity avoidance [43].

Lastly, we note that the prototype used in the experimental validation of this work was designed as a proof of concept without a specific application in mind. Future work should focus on designing TDPCR structures tailored to specific application scenarios, for which the proposed kinetostatic modeling framework can serve as a foundation.

In conclusion, this paper extends the current state-of-the-art, as it proposes a comprehensive modeling and evaluation framework for TDPCR. We envision that these novel manipulators will ultimately enable new application areas, in which multiple coupled tendon-driven continuum robots are operating in highly constrained environments. We believe, this work will be fundamental for both investigating and synthesizing optimized TDPCR designs with respect to specific application scenarios and desired kinematic properties as well as controlling the resulting robotic structures.

APPENDIX A
MULTIMEDIA APPENDIX

The appendix includes a video showcasing motion examples of the proposed TDPCR, an animated visualization of the three-dimensional workspaces, manipulability and compliance ellipsoids in different configurations as well as a side-by-side comparison of the robot prototype and simulations.

REFERENCES

[1] C. E. Bryson and D. C. Rucker, "Toward parallel continuum manipulators," in IEEE International Conference on Robotics and Automation, 2014, pp. 778–785.

[2] A. L. Orekhov, C. B. Black, J. Till, S. Chung, and D. C. Rucker, "Analysis and validation of a teleoperated surgical parallel continuum manipulator," IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 828–835, 2016.

[3] B. Mauzé, G. J. Laurent, R. Dahmouche, and C. Clévy, "Micrometer positioning accuracy with a planar parallel continuum robot," Frontiers in Robotics and AI, vol. 8, p. 196, 2021.

[4] E. M. Young and K. J. Kuchenbecker, "Implementation of a 6-dof parallel continuum manipulator for delivering fingertip tactile cues," IEEE Transactions on Haptics, vol. 12, no. 3, pp. 295–306, 2019.

[5] O. Altuzarra and J. P. Merlet, "Certified kinematics solution of 2-dof planar parallel continuum mechanisms," in IFToMM World Congress on Mechanism and Machine Science. Springer, 2019, pp. 197–208.

[6] B. Mauze, R. Dahmouche, G. J. Laurent, A. N. Andre, P. Rougeot, P. Sandoz, and C. Clevy, "Nanometer Precision with a Planar Parallel Continuum Robot," IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3806–3813, 2020.

[7] Z. Yang, X. Zhu, and K. Xu, "Continuum delta robot: a novel translational parallel robot with continuum joints," in IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2018, pp. 748–755.

[8] A. W. Mahoney, P. L. Anderson, P. J. Swaney, F. Maldonado, and R. J. Webster, "Reconfigurable parallel continuum robots for incisionless surgery," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016, pp. 4330–4336.

[9] C. B. Black, J. Till, and D. C. Rucker, "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing," IEEE Transactions on Robotics, vol. 34, no. 1, pp. 29–47, 2018.

[10] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," The International Journal of Robotics Research, vol. 38, no. 6, pp. 723–746, 2019.

[11] F. Zaccaria, S. Briot, M. T. Chikhaoui, E. Idá, and M. Carricato, "An analytical formulation for the geometrico-static problem of continuum planar parallel robots," in Symposium on Robot Design, Dynamics and Control. Springer, 2020, pp. 512–520.

[12] G. Chen, Y. Kang, Z. Liang, Z. Zhang, and H. Wang, "Kinetostatics modeling and analysis of parallel continuum manipulators," Mechanism and Machine Theory, vol. 163, p. 104380, 2021.

[13] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, "Efficient computation of multiple coupled Cosserat rod models for real-time simulation and control of parallel continuum manipulators," in IEEE International Conference on Robotics and Automation, 2015, pp. 5067–5074.

[14] S. Briot and A. Goldsztejn, "Singularity Conditions for Continuum Parallel Robots," IEEE Transactions on Robotics, pp. 1–19, 2021.

[15] J. Till and D. C. Rucker, "Elastic stability of cosserat rods and parallel continuum robots," IEEE Transactions on Robotics, vol. 33, no. 3, pp. 718–733, 2017.

[16] O. Altuzarra and F. J. Campa, "On singularity and instability in a planar parallel continuum mechanism," in International Symposium on Advances in Robot Kinematics, 2020, pp. 327–334.

[17] D. Díaz-Caneja, F. J. Campa, and O. Altuzarra, "Design and modeling of a parallel continuum manipulator for trunk motion rehabilitation," Journal of Medical Devices, vol. 15, no. 1, p. 011109, 2021.

[18] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum Robots for Medical Applications: A Survey," IEEE Transactions on Robotics, vol. 31, no. 6, pp. 1261–1280, 2015.

[19] M. Russo, N. Sriratanasak, W. Ba, X. Dong, A. Mohammad, and D. Axinte, "Cooperative continuum robots: Enhancing individual continuum arms by reconfiguring into a parallel manipulator," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 1558–1565, 2022.

[20] K. Nuelle, T. Sterneck, S. Lilge, D. Xiong, J. Burgner-Kahrs, and T. Ortmaier, "Modeling, calibration, and evaluation of a tendon-actuated planar parallel continuum robot," IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 5811–5818, 2020.

[21] S. Lilge, K. Nuelle, G. Boettcher, S. Spindeldreier, and J. Burgner-Kahrs, "Tendon actuated continuous structures in planar parallel robots: A kinematic analysis," Journal of Mechanisms and Robotics, vol. 13, no. 1, 2021.

[22] G. Böttcher, S. Lilge, and J. Burgner-Kahrs, "Design of a reconfigurable parallel continuum robot with tendon-actuated kinematic chains," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1272–1279, 2021.

[23] A. Lotfavar, S. Hasanzadeh, and F. Janabi-Sharifi, "Cooperative continuum robots: concept, modeling, and workspace analysis," IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 426–433, 2017.

[24] F. Janabi-Sharifi, A. Jalali, and I. D. Walker, "Cosserat rod-based dynamic modeling of tendon-driven continuum robots: A tutorial," IEEE Access, vol. 9, pp. 68 703–68 719, 2021.

[25] A. Jalali and F. Janabi-Sharifi, "Dynamic modeling of tendon-driven comanipulative continuum robots," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 1643–1650, 2021.

[26] D. C. Rucker and R. J. W. III, "Statics and dynamics of continuum robots with general tendon routing and external loading," IEEE Transactions on Robotics, vol. 27, no. 6, pp. 1033–1044, 2011.

[27] E. Amanov, T.-D. Nguyen, and J. Burgner-Kahrs, "Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions," The International Journal of Robotics Research, vol. 40, no. 1, pp. 7–23, 2021.

[28] K. Oliver-Butler, J. Till, and C. Rucker, "Continuum robot stiffness under external loads and prescribed tendon displacements," IEEE Transactions on Robotics, vol. 35, no. 2, pp. 403–419, 2019.

[29] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," Frontiers in Robotics and AI, vol. 7, p. 223, 2021.

[30] M. T. Chikhaoui, S. Lilge, S. Kleinschmidt, and J. Burgner-Kahrs, "Comparison of modeling approaches for a tendon actuated continuum robot with three extensible segments," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 989–996, 2019.

[31] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in IEEE International Conference on Robotics and Automation, 2011, pp. 945–950.

[32] D. Zlatanov, R. G. Fenton, and B. Benhabib, "Singularity analysis of mechanisms and robots via a motion-space model of the instantaneous kinematics," in IEEE International Conference on Robotics and Automation, 1994, pp. 980–985.

[33] M. Galassi et al., "GNU scientific library reference manual," 2018. [Online]. Available: https://www.gnu.org/software/gsl/

[34] J. Burgner-Kahrs, H. B. Gilbert, J. Granna, P. J. Swaney, and R. J. Webster, "Workspace characterization for concentric tube continuum robots," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 1269–1275.

[35] W. Amehri, G. Zheng, and A. Kruszewski, "Workspace boundary estimation for soft manipulators using a continuation approach," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 7169–7176, 2021.

[36] A. Walid, G. Zheng, A. Kruszewski, and F. Renda, "Discrete cosserat method for soft manipulators workspace estimation: An optimization-based approach," ASME Journal of Mechanisms and Robotics, vol. 14, no. 1, 2022.

[37] J.-P. Merlet, Parallel robots. Springer Science & Business Media, 2006, vol. 128.

[38] F. Zaccaria, E. Idá, S. Briot, and M. Carricato, "Workspace computation of planar continuum parallel robots," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 2700–2707, 2022.

[39] T. Yoshikawa, Foundations of robotics: analysis and control. MIT press, 1990.

[40] V. Aloi, C. Black, and C. Rucker, "Stiffness control of parallel continuum robots," in ASME 2018 Dynamic Systems and Control Conference, 2018.

[41] R. J. Roesthuis and S. Misra, "Steering of multisegment continuum manipulators using rigid-link modeling and FBG-based shape sensing," IEEE Transactions on Robotics, vol. 32, no. 2, pp. 372–382, 2016.

[42] S. Sabetian, T. Looi, E. D. Diller, and J. Drake, "Self-collision detection and avoidance for dual-arm concentric tube robots," IEEE Robotics and Automation Letters, pp. 1–1, 2019.

[43] J. Kotlarski, B. Heimann, and T. Ortmaier, "Influence of kinematic redundancy on the singularity-free workspace of parallel kinematic machines," Frontiers of Mechanical Engineering, vol. 7, no. 2, pp. 120–134, 2012.

Sven Lilge received a B.Sc. and a M.Sc. degree in electrical engineering and information technology from the Leibniz University Hannover, Germany in 2014 and 2016, respectively. From 2017 to 2019 he worked as a research assistant at the Laboratory for Continuum Robotics at the Leibniz University Hannover. He is Ph.D. Candidate with the Department of Computer Science and a research assistant with the Continuum Robotics Laboratory at the University of Toronto, Canada. His research is primarily concerned with parallel continuum robots focusing on design, modeling, control and kinematic evaluation.

Jessica Burgner-Kahrs is an Associate Professor with the Departments for Mathematical & Computational Sciences, Computer Science, and Mechanical & Industrial Engineering, the founding Director of the Continuum Robotics Laboratory (CRL), and Associate Director of the Robotics Institute at the University of Toronto, Canada. From 2013 to 2019 she was with Leibniz University Hannover, Germany and from 2010 to 2012 with Vanderbilt University, USA. She received her Diploma and Ph.D. in computer science from Karlsruhe Institute of Technology (KIT), Germany in 2006 and 2010 respectively. Her research focus lies on continuum robotics and in particular on their design, modelling, planning and control, as well as human-robot interaction. Her fundamental robotics research is driven by applications in minimally-invasive surgery and maintenance, repair, and operations.